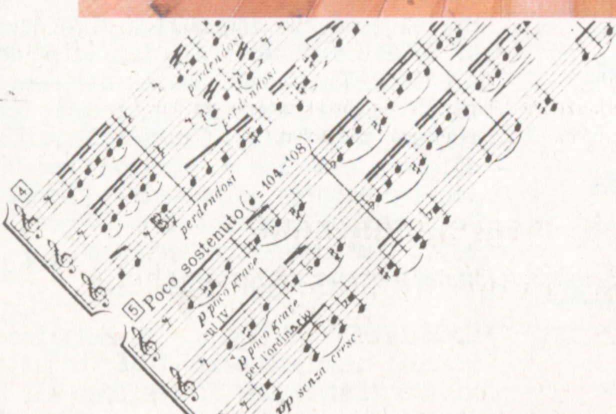SPECTRUM

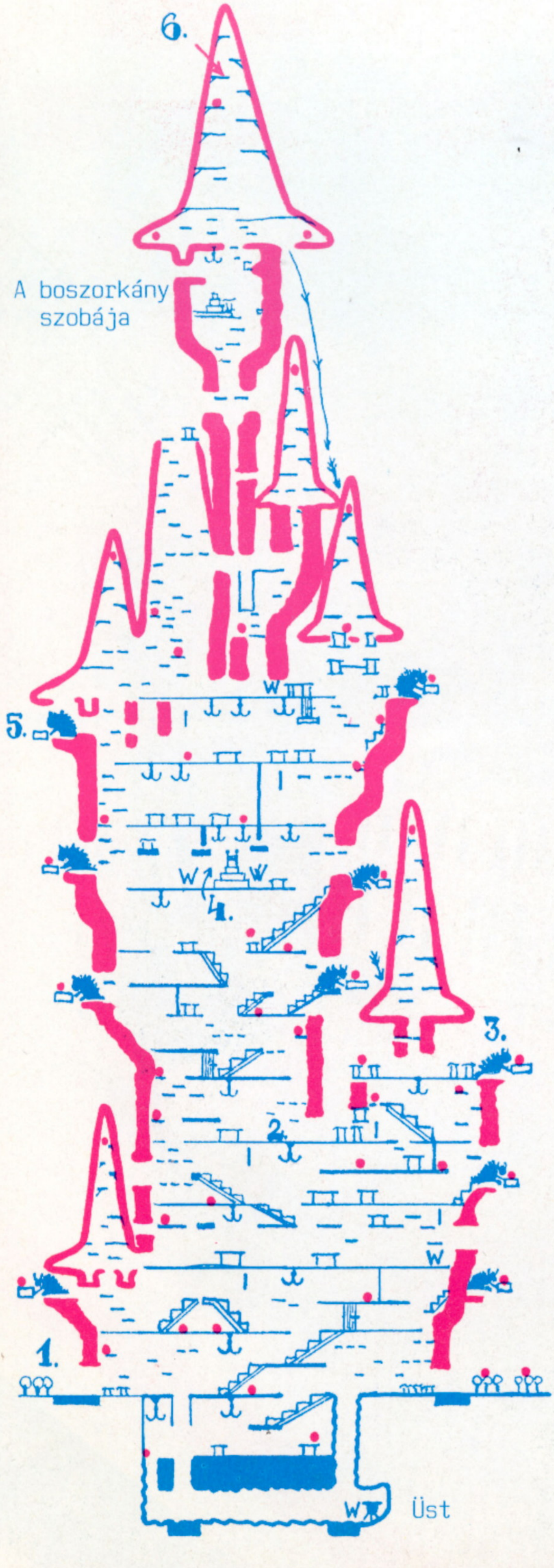
VILÁG V. rész







Cauldron 2.



A piac igényeinek megfelelően a PALACE cég 1986-ban megjelentette CAULDRON című programjának II. részét. Igazán mélyreható változások nem történtek a játék meséjében az első részhez képest: egy elvarázsolt ingatlanban kell - mindenféle nyaktörő mutatványok során - különféle machinációkat elkövetnünk, hogy a rút főbérlőt (egy boszorkányt) megfoszthassuk hatalmától.

Nem is szaporítjuk tovább a szót, vágjunk rögtön a dolgok sűrűjébe. Mint az már bizonyára mindenkinek feltűnt, ez itt balra egy - kissé "stilizált" - térkép, ami a játék folyamán nélkülözhetetlen segítőtársunk lesz, jelölései alapján magyarázzuk el a játék lényegét.

A kastélyban 6 felszerelési tárgyat kell összeszednünk, amelyek a térképen nagy számokkal vannak jelölve:

1. scissors (olló)	2. stick (fejsze)
3. glass (üvegkehely)	4. crown (korona)
5. status	6. book (könyv)

Miközben ezeket szedegetjük, meg kell szereznünk a kastély - egyben boszorkány kartársnő - mágikus erejét eredményező varázstablettákat. Ezekből van néhány (úgy kb. 50) darab, általában a legjobb helyeken elszórva (pl. a tornyok tetején vagy a sárkányok előtt), a térképen sötét ponttal (" ") jelöltük őket.

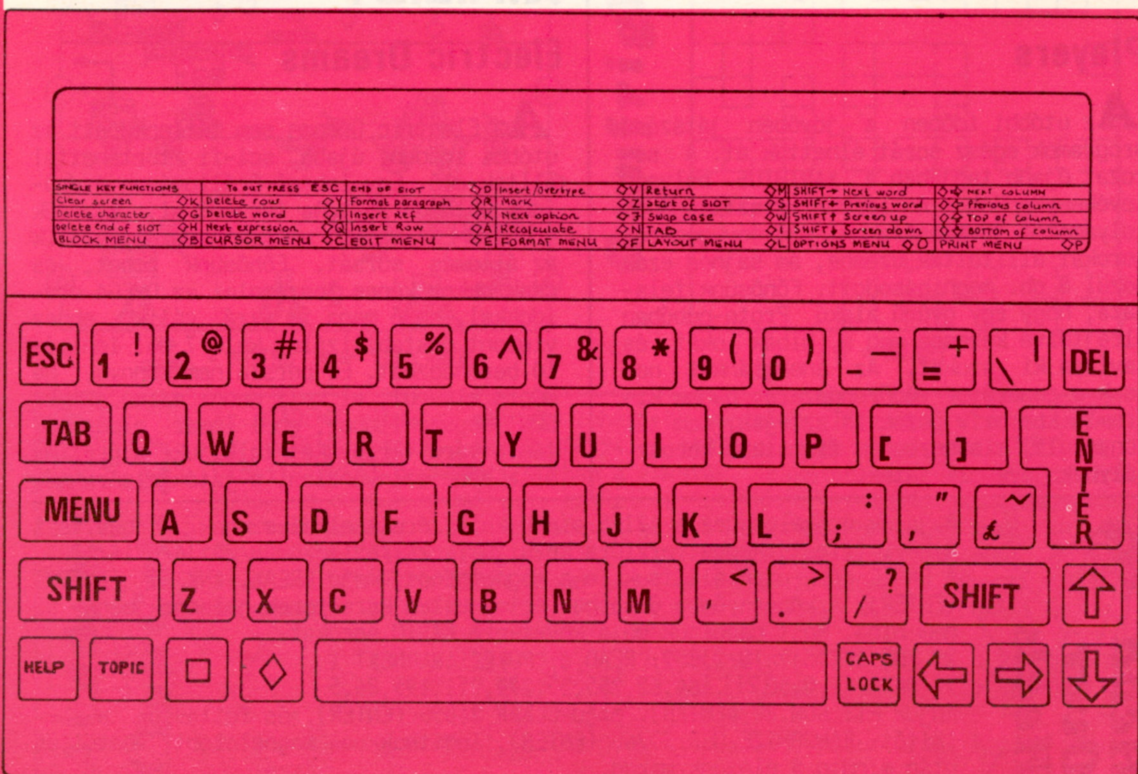
Mászkálásunkat boszorkány kartársnő nem nézi olyan túláradó örömmel: némi konkurrenciát, varázslókat (WIZARD) és különböző ízeltlábúakat (INSECT) küld ellenünk. Ezek a térképen 'W' illetve 'I' betűvel vannak jelölve.

Ha felcuccoltuk magunkat a 6 tárggyal és az ÖSSZES varázsporral, fel kell mennünk a boszorkány budoárjába és ott némi fodrászati divatbemutató keretében le kell vágnunk a hajzatát. Ezután már nincs sok dolgunk, csak ide a közelbe megyünk: a legalsó szinten elhelyezett üstbe bele kell dobnunk a boszorkány levágott szőrzetét (mindenesetre érdekes egy leves készül ebben az üstben).

ÖrökéletPOKE: 52974,0. A Multiface-szel készült másolatokhoz a POKE bejuttatását kiváló kiadványunk 3. részében már közöltük. A többieknek kellemes védelemfeltörést kívánunk!

1. 1987/Z-88

A II.részben már említettünk néhány szót Sinclair új 'csodájáról' a Z-88-ról. Nos a gép első szériája azóta már kifutott a piacra, bár a hozzáfűzött reményeket a mai napig nem váltotta be, Sir Clive bizakodik, ha nem is lett a Z-88 1987 gépe, akkor majd nevéhez méltóan talán 1988-ban. Az ábrán is látható gép mérete a gépírópapírnak megfelelő, s lapos, mint a hagyományos 48K-s Spectrum. Billentyűzetkiosztása átmenet a Spectrum és az IBM között, két új szimbólum is szerepel, a '□' és a '◇', melyek speciális funkciókat hivatottak elősegíteni. A Supertwist LCD kijelzőn 80 oszlopban 8 sor információt tekinthetünk át.
A gép memóriája: 128K ROM, amely magába foglalja az op. rendszert, valamint a beépített software-eket (egy szövegszerkesztőt, egy táblázatkezelőt és egy adatbáziskezelőt). A beépített rezidens 32K RAM-ból közvetlenül 15K áll rendelkezésre, de ez elég kb. 2000 szó beviteléhez. Természetesen a RAM bővíthető, a gép első részén 3 db. cartridge csatlakozó helyet alakítottak ki. 32K/128K/1Mbyte cartridge-ok állnak rendelkezésre. Ez utóbbi elég drága, így javasolt a 128K-s cartridge-ok felhasználása. Ezzel a módszerrel már 416K RAM memóriánk van, ami elég kb. 200 oldal szöveg eltárolásához. A CMOS technológia lehetővé teszi, hogy a beépített kondenzátorok támogatásával arra az időre se veszszen el a RAM tartalom, amíg elemet cserélünk, ugyanis a Z-88 4 db. 'AA' ceruza elemmel működik. Bekapcsolt állapotban kb. 20 órás üzemeltetés után ajánlatos elemet cserélni, kikapcsolt állapotban akár 1 évig is rendelkezésre áll a RAM tartalom.
Amíg más számítógép háttértárolóként magnetofont, vagy lemezegységet használ, addig a Z-88 cartridge-eket. Felvetődik a kérdés, akkor mégis miért hasznos a gép alkalmazása. Nos a Z-88 valóban átmenet a kisszámítógépek és az IBM konfigurációk között. A Z-88 által tárolt adatok mindegyike problémamentesen áttölthető RS-232 vonalon IBM kompatibilis számítógépeken futó valamely rendszer-sofware-be. A cél-software a konvertáláshoz kiválasztható, a konvertáló software-t pedig a gyártó külön 5 1/4"-os lemezen hozta forgalomba IBM és kompatibilis gépekre. A jelenlegi konvertáló software kompatibilis a Word Starral, Word Perfect-tel és a LOTUS I-II-III verziókkal is. Fontos megemlíteni, hogy a beépített modem leilleszthető bármilyen szabványos telefonvonalra.
Megállapíthatjuk tehát, hogy az új 'SINCLAIR gép' elsősorban a megrögzött IBM felhasználók földön-vízen-levegőben jelentkező ujj-bizsergését célozza enyhíteni.



PHANTOM CLUB

Ocean

A program jugoszláv készítőjének - Dusko Dimitrijevicnek - az első bemutatkozása az IMAGINE címke alatt megjelent 'MOVIE' c. program volt. Most is hasonló kivitelű, színvonalas játékkal bizonyított. A Phantom Club tele van bűnözőkkel, vezérük Zarg, akire hallgat az egész banda. A klubban törzsvendég Plutus is, aki még jófiú, és elhatározza, hogy felgöngyölíti a piszkos ügyeket. Plutus bolyong a Phantom Club labirintusaiban - amely 550 szobából áll -, és keresi a banda főhadiszállását, hogy leszámoljon Zarg-gal. Az egyes szobákban barátokkal és ellenségekkel is találkozik - ez utóbbiak sokszor különös formát öltenek (szerzetes, pókember, stb.) -, fel kell mérni a helyzetet, ha ellenség, tüzelni kell. Az itt ott található gömbök, gyémántok extra energiát, extra sebességet, extra BONUS-t eredményeznek. A MOVIE-hoz és BAT-MAN-hez hasonló grafikájú játék izgalmas kalandot jelent mindannyiunk számára.

ACE-2

Cascade Games

Hadseregünk kémrepülőgépe behatolt az ellenséges területre, aminek az lett a következménye, hogy az ellenség vadászgépet indított a levegőbe, a kémrepülőgépünk megsemmisítésére. Az ACE feladata, hogy megvédje társát, és megsemmisítse az ellenséget. A játék elődjéhez hasonlóan kiváló grafikával rendelkezik, műszerfalán megtalálható a radar, iránytű, orrszög és forgás-indikátor, magasság műszer. A navigáció nagyszerűen kivitelezhető. Lehetőség van a motor-fordulat finomszabályozására, az egyes motorok külön külön ki-/bekapcsolhatók (pl. tűz esetén). A gép a 60000 láb csúcsmagasságot is elérheti. Lőszerünk bőven akad, a normál géppuskán kívül légi gépágyúval, hőérzékelő vagy radarirányítású rakétákkal, ezen belül lég-föld vagy lég-lég rakétákkal is rendelkezünk. A meghibásodott gép a start-anyahajóra visszaszállva kijavítható. A Cascade Games ismét nagyszerű szimulációval lepte meg a piacot.

XANTHIUS

Players

Az utóbbi időben a jövőben játszódó programok egész sorát élhettük át. A sorozat újabb tagjában a XANTHIUS bolygón tevékenykedünk. A légkör tisztaságának folyamatos javítására un. növény-generátorokat állítottak üzembe, ám sajnos ezek közül 8 db. meghibásodott. Főhősünk feladata, hogy egy gyűrű alakú Proto-gubóban elrejtőzve keresse meg a hibák okát és javítsa ki azokat. Az energiánkat nem csak a generátor védőrendszere, hanem kiáramló korrozív gázok is szívják. Nem túl bonyolult, elsősorban a fiatalabb korosztálynak javasolt játék.

NIHILIST

Electric Dreams

Az Electric Dreams sem tétlenkedett az utóbbi időben, újabb sci-fi történettel álltak elő. Egy lebegő űrállomáson a droidok különös fajtája terjedt el, közülük 4 un. OMEGA droid, melyek veszélyeztetik az állomás sorsát. Főhősünk robot, és csak akkor képes leszámolni az OMEGA droidokkal, ha ő maga is azzá válik, ehhez viszont destabilizálni kell 7 db. ionizáló berendezést. Ez persze nem könnyű, ha mégis sikerül, el kell menni az un. 'MESON' megszakítóhoz, ahol immár elpusztításra váró társainkat találjuk. A grafika és a zene szinvonalas, a játék elsőrangú.

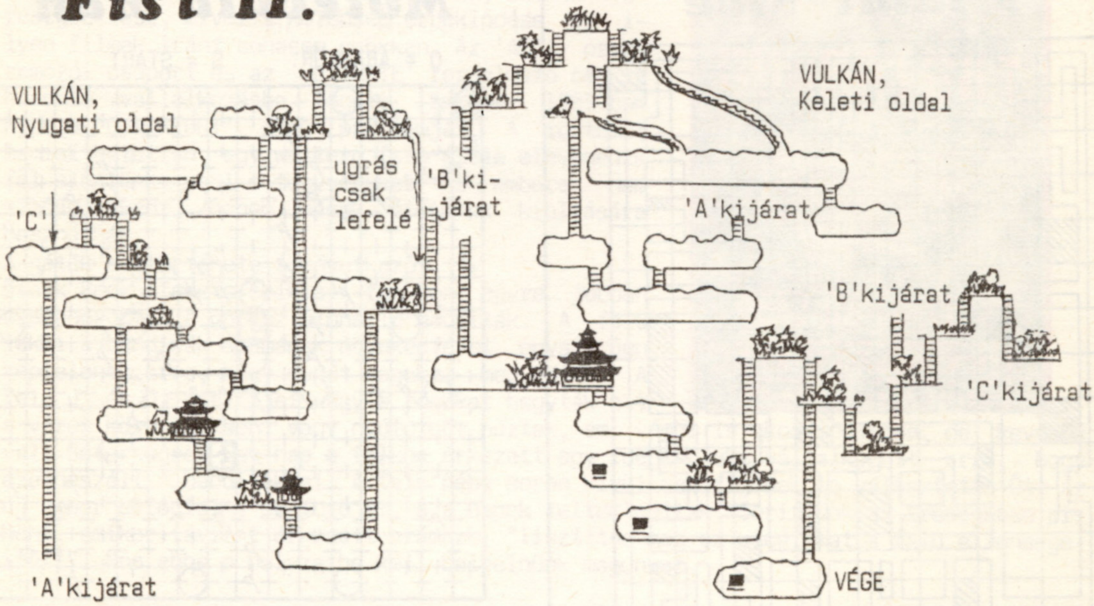


A játék örökéletéhez a 42195.címen kell zérust elhelyeznünk. A bevitelt a köv. file-térképes verzióra közöljük: 241/6913/20000/20536/6916/1704. A BASIC rész után állítsuk le a magnetofont, majd BREAK, INK 7 (ENTER) (ENTER). A képernyőn a 0.sorban értelmetlen karakterek jelennek meg, ez nem hiba. Most gépeljük be a következő BASIC sorokat:

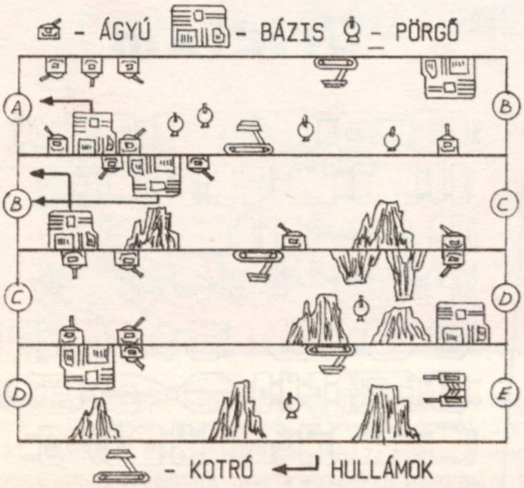
```
20 FOR i=23847 TO 23859: READ a: POKE i,a: NEXT i
30 DATA 62,0,50,211,164,62,255,55,229,251,195,86,5
```

Adjuk ki: RUN 20 (ENTER), majd CLEAR 24899 (ENTER), és futtassuk a betöltőt: RANDOMIZE USR 23760 (ENTER). Indítsuk a magnetofont, betöltés után mindig 8 életünk marad.

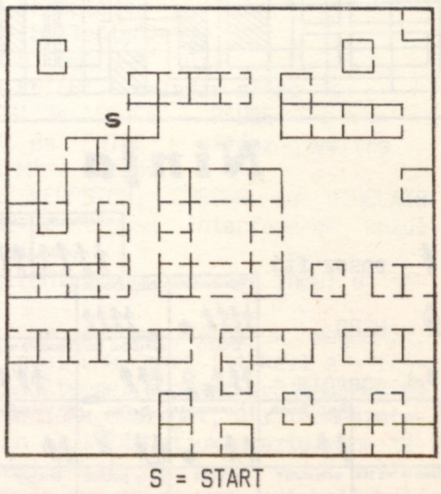
Fist II.



Transmuter



Gunfright



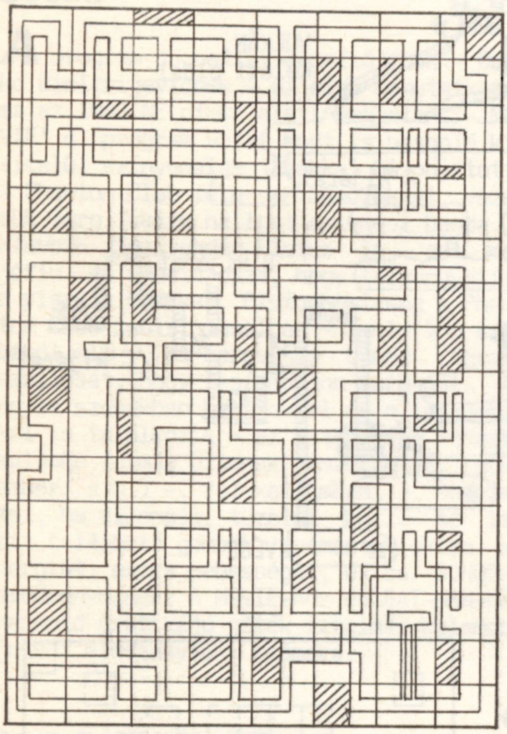
Academy (Tau Ceti 2)

Az ACADEMY játékban előnyös lehet, ha végtelen lőszerrel rendelkezünk. Az ACADEMY a 44716, 44752, 44794 és 44831 címeken elhelyezett 'DEC (HL)' utasításokkal csökkenti sorban a rakéták, rakétaelhárítók, világítórakéták és bombák számát. Ha ezeket az értékeket kinullázzuk, lőszereink nem csökkennek. A bevitel módját arra a verzióra közöljük, amelynek file-térképe: 548/6942/40168/536/...; Tekerjük a kazettát oda, ahol a 40168-as blokk kezdődik, majd gépeljük be és futtassuk a következő BASIC programot:

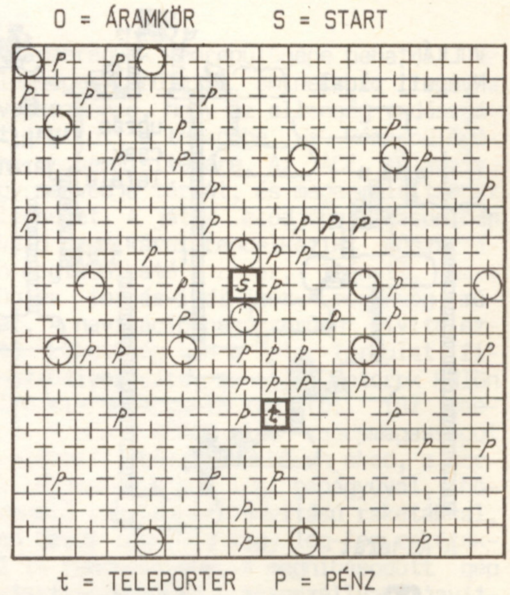
```
10 CLEAR 24831: FOR i=23296 TO 23341: READ a: POKE i,a:
   NEXT i: RANDOMIZE USR 23296
20 DATA 221,33,0,97,17,232,156,62,31,55,205,86,5,221,33,232,
   253,17,24,2,62,37,55,205,86,5,62,0,50,172,174,50,208,174,
   50,250,174,50,31,175,195,0,97
```

Indítsuk a magnetofont, betöltődés után a lőszereink száma mindig állandó lesz!

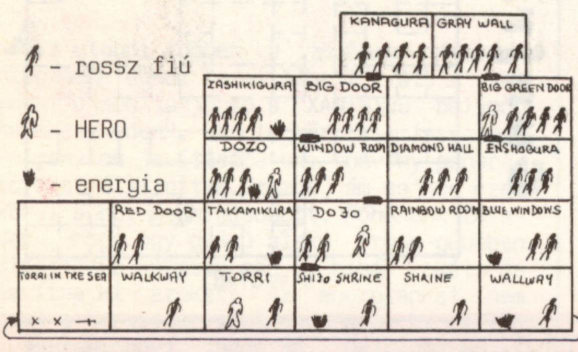
Miami Vice



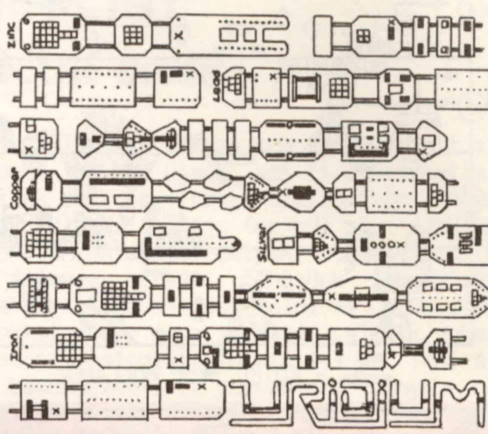
Molecula Man



Ninja



Uridium



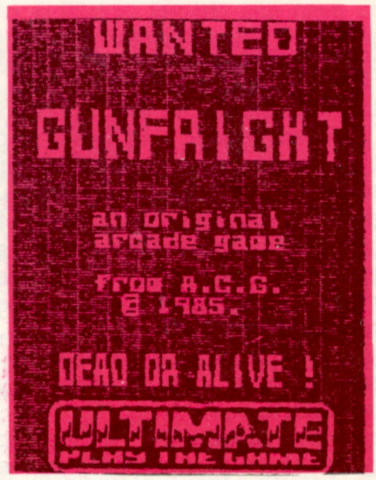
Star Raiders II.

A végtelen energiához a 37806-os címre 24-et, a 37807-es címre pedig 250-et kell beírnunk. A mi verziónk file-térképe: 626/100/778/256/30000. A bevitelhez tegyük a következőket: A BASIC rész betöltése után BREAK, majd állítsuk le a magnetofont. Írjuk be: POKE 23570,6 (ENTER), ezután listázzuk ki a program 210-es sorát. Töröljük a végéről a RANDOMIZE USR 65024 utasítást, és gépeljük be a következő BASIC sorokat:

```
220 FOR i=65091 TO 65104: READ a: POKE i,a: NEXT i
230 DATA 62,24,50,174,147,62,250,50,175,147,251,195,48,117
240 RANDOMIZE USR 65024
```

Adjuk ki: RUN (ENTER) és indítsuk a magnetofont. Betöltés és start után, ha a bal-felső műszeren elfogy az energiánk, a játék folytatódik.

2.1 GUNFRIGHT



Amikor egy filmgyárban western-film forgatásába kezdenek, akkor már előre tudják, hogy filmük nyereséges lesz, mivel a közönség érdeklődése az ilyen filmek iránt sohasem csökken. Az 'ACG' programozói csoport és az 'ULTIMATE' forgalmazó cég is hasonló vállalkozásba fogott, amikor 1985-ben kiadta 'GUNFRIGHT' című programját. A történet hamisíthatatlan, egy western falu élete elevenedik fel előttünk, ahol a fegyvereket az emberek nem szobadíszként, hanem mások életének kioltására használják.

A játék előtörténete a következő:

Black Rock (fekete szikla) faluban egyre jobban elszaporodtak a gonosz útonálló banditák. A falu lakói a banditák számának növekedtével egyszerűen képtelenek arra, hogy saját magukat megvédjék. A falu új és új seriffjei vagy a fogukat hagyták ott a véres küzdelmekben, vagy nyúlcipőt húztak, ami végül is okos megoldás, de kevésbé vall bátorságra. Egy nap a faluba érkezett egy idegen, aki vállalkozott arra, hogy szembeszáll a banditákkal. A falu népe soron kívül seriffjének is választotta Őt. Az új ember elfoglalta pozícióját, s a napok teltek múltak. Seriffünk az irodájában pihent, amikor távirat érkezett számára: "Tisztítsd meg az utcáinkat a falu ellenségeitől!". Nos ebbe a helyzetbe kell beleélnünk magunkat.



A főmenüben beállíthatjuk a játékosok számát (1 játékos/2 játékos). Kiválaszthatjuk a vezérlést:

KEYBOARD (billentyűzet),

'1-0' és 'Q-P' tűz
'A-ENTER' - Haladás előre
'CS' és 'SPACE' - PAUSE
'Z' és 'ENTER' - látószög váltás
'X-M' - forgás

valamint KEMPSTON, CURSOR és SINCLAIR INTERFACE 2 típusú Interface-ek közül választhatunk.

A '3' billentyű megnyomására indul a játék.

Hirtelen pénzesőben találjuk magunkat. Nagyon figyeljünk, mert itt kell a 'fejvadászat'költségeire pénzt gyűjtenünk. A képernyő alján figyelemmel kísérhetjük megszerzett pénzünk összegét, ami természetesen a program jóvoltából 250 dollárról indul. Szintén alul láthatjuk pisztolyunk tárát, amiben sárga karikák jelzik a még meglévő töltényeinket.

A pénzes erszények pottyanásakor megjelenik a képernyőn egy célzó kör, ezt kell bal/jobb/fel/le mozgatással az erszényekre mozgatni, és tüzeléssel tehetjük magunkévá ezeket az erszényeket. Itt a billentyű-vezérlés a következő:

'1-0' - tűz ; 'Q-P' - fel
'A-ENTER' - le ; 'X,V,B' - balra
'C,N' - jobbra

A pénzeső elmúltával seriffünket láthatjuk, amint a börtön előtt álldogál. Bal oldalon egy nagy falragasz képe látható, melyen a körözött személy neve, fényképe, és a fejére kitűzött díj lett feltüntetve. Alatta a kis kalapkák életeink számát jelzik, míg lejjebb pénzünk fogyását kísérhetjük figyelemmel. A képernyő alján egy távirat (TELEGRAM) is látható. Ezen a töltények (BULLETS), lovak (HORSE) és a büntetés (FINE) pénzbeli értékét láthatjuk. Érdemes erre odafigyelnünk, mert ha a pénzünk elfogy, - töltény híján - nem fogunk tudni megküzdeni ellenségeinkkel. Ezért tehát takarékoskodjunk töltényünkkel, és ne ragaszkodjunk feltételenül a költséges lovagláshoz. Fokozottabb takarékoskodás végett még arra is figyelhetünk, hogy pisztolyunkat akkor töltsük, amikor a töltény ára éppen minimumon van (vagyis 5 dolláron). A tár

2.1 GUNFRIGHT

automatikusan töltődik, ha kilőttük az utolsó lövedéket. Az imént említett taktika vonatkozik a lovak használatára is, mivel itt akár 50 dollárt is kitehet az árak közötti különbség. Ha mégis valami ok folytán elfogyna a pénzünk, még ne essünk pánikba. Szerencsénkre a falucska a pénzszűkében lévő embereket tiszteletreméltóan támogatja, ebben az esetben akár 'órákat' is lovagolhatunk. Pisztolyunk tárának feltöltése is könnyen megoldható. Az utca 'porában' itt-ott néhány erszényt találunk. Ha ezeket felvesszük, pisztolyunk tára ismét feltöltődik.

Legfontosabb feladatunk persze a bandita kézrekerítése. Ehhez az akcióhoz a térkép segítségével kezdhetünk hozzá, de vigyázzunk, mert a falucska utcái sok meglepetést tartogatnak számunkra. Black Rock tele van rohanó asszonyokkal, akik kis kosárkával a kezükön lótnak-futnak fel-alá. Seriffünket próbáljuk megóvni a hölgyektől, mivel a velük való találkozás életünkbe kerül, sőt még büntetést is kell fizetnünk. 'Elhalálozásunk' után - ha még nem fogytak el az életeink - a játék ismét a börtön elől folytatódik.

Esetleges haragunkat - lehetőség szerint - ne az asszonyok és gyermekek lelövésével vagy lóval való elgázolásával vezessük le, mert így zsebünk - a büntetések miatt - hamar kiürülhet.

Az utcákon lovak is találhatók. Ezek az amúgy nem túl gyors tempónkat jelentősen felgyorsítják, igaz elég tekintélyes összeg ellenében. Lovaskalandunk rövid idő elteltével véget ér, mivel a pénzünkért csak meghatározott ideig lovagolhatunk, így a keresést újból 'lábon' folytathatjuk. Ha lovaglás közben mégis elkap minket a láz, ne feledjük, az út szélén található kaktuszokat nem tudjuk az asszonyokhoz hasonlóan legázolni, sajnos a kaktuszok tüskéinek szúrása mindig halálos kimenetelű.

Szerencsére időbeli korlátaink nincsenek, tehát - ha kedvünk tartja -, megtehetjük, hogy a térkép segítségével először is megismerkedünk a faluval. Ha már sikerült felfedezni a falucska utcáinak rejtelmeit, akkor már tényleg elkezdhetjük a 'fejvadászatot'. A keresés persze egy szál térképpel meglehetősen hosszú és nehézkes lenne, de szerencsénkre vannak, akik a seriffünk oldalán állnak, ezek pedig a falu ifjú fiai. Az utcákon sok helyen találkozhatunk ezekkel a fiúkkal, ahogy fel/le ugrálva - nem kis feltűnést keltve -, pisztolyukkal mutatják azt az irányt, amerre a bandita nyomára akadhatunk. Az ifjak nem a pontos irányt mutatják, mivel ellenfelünk is mozog, de - ha a mutatott irányban újabb ifjút keresünk, akkor előbb utóbb eljutunk a keresett helyre, ahol megpillantjuk a körözött egyént. Ügyeljünk arra, hogy ne veszítsük szem elől, de ne is menjünk túl közel hozzá, mert ez is életünk elvesztésével járhat - és különben is felesleges, mert nem megverekedni kell vele. A megfelelő pillanatban egy jól irányzott lövéssel terítsük földre a banditát. Ezzel persze még nincs vége az akciónak, mivel seriffünk a sportszerűség nagymestere, így csak a lábába lőtt, hogy később egyenlő esélyekkel megküzdhessen ellenfelével. Ekkor jön a párbaj. Itt a bandita 'nagyalakú' képét láthatjuk, amely szemünknek elég tetszetős látványt nyújt, de nehogy elfeledkezzünk igazi feladatunkról, a párbajról.

Célkeresztünket a lehető leggyorsabban irányítsuk a bandita testére, ezután pedig lőjünk eszeveszett gyorsasággal. Mivel minden párbaj kétesélyes, itt is csak egy győztes lehet. Ha a bandita elesik, és csizmájának a talpát mutogatja (azaz 'feldobta a talpát'), akkor örömmel tapsikolhatunk, mivel mi győztünk. Viszont, ha a képernyőn kerek lövés-nyomok láthatók, akkor csendes felállással tisztelegjünk volt seriffünk holtteste előtt.

Ha az első variáció történik meg - tehát mi nyertünk -, akkor készülhetünk a következő bandita elkapására. A legügyesebbeknek 10 banditával kell megküzdeniük, sorrendben a következőkkel: Buffalo Bill, Billy the Kid, Sundance Kid, Jesse Jones, Butch Cassidy, Ma Baker, Wild Bill, Rumpo Kid, F.Clever Jake, Millery Bar Kid.

A jól dolgozó seriffek jutalomban részesülnek. Ez abban nyilvánul meg, hogy a Jesse Jones feletti győzelem után újabb pénzeső következik, de itt már az erszények gyorsabban pottyannak. Ugyanez történik Rumpo Kid után is, természetesen még gyorsabb erszényekkel.

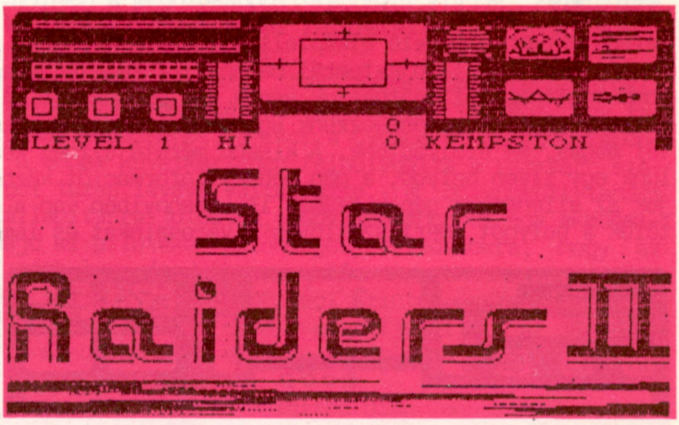
E program igazán megérdemel minden elismerést, mert izgalma, változatossága, története, grafikája sok-sok játékon túltesz. Aki egyszer elkezdi a 'fejvadászatot', biztos, hogy nem nagyon tud utána elszakadni a játéktól. Az 'ULTIMATE' a maga idejében ezzel a programjával 'nagyot alakított'.

2.2 STAR RAIDERS II.

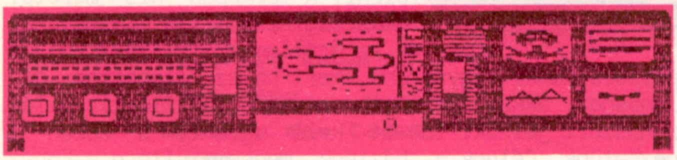


Az ELECTRIC DREAMS cég neve csak 1985 óta vésődött be a Spectrumosok tudatába. Az első igazi nagy dobás az 'I of the mask' c. program volt, melynek szerzői, a 'SPACEMAN' programozói csoport a 'Quicksilva' cégtől pártoltak át, olyan nagyszerű alkotások után, mint az 'ANT ATTACK', vagy a 'ZOMBIE-ZOMBIE'. Nos az azóta eltelt két év alatt az ELECTRIC DREAMS kitett magáért, több kiemelkedő software-t forgalmaztak, pl. az Ő kezük alól került ki a forgalomba a HIJACK, a PRODIGY, az ENDURO RACER, a SUPER-SPRINT, de a BIG TRBL. IN LITTLE CHINA is. Az 1986-ban elkészült STAR RAIDERS c. akció játék olyan nagy sikert aratott, hogy folytatását is piacra dobták STAR RAIDERS II. néven, 1987-ben. A játék alapötlete először ATARI játékautomatára készült el, s csak ezután dolgozták fel személyi számítógépekre.

A történet 2050-ben játszódik, főhősünk - a játékos - különleges feladatokra lett kiképezve, s egyszemélyes szuper-űrsiklójával meg kell védenie a bolygórendszert a szomszédos, egyben ellenséges bolygórendszer un. 'ZYLON' flottáinak támadásaitól. Elsődleges feladatunk a védekezés, amennyiben helyzeti előnyt szerzünk, akár támadhatunk is.



A játék bejelentkezésekor a képernyő felső részén megjelenik az űrsikló műszerfala, alul a start-bolygó felületének részlete látható. A játék kezdetén be kell állítanunk a vezérlést, és a nehézségi szintet. Állítsuk be előbb a vezérlést. A bejelentkezéskor a műszerfal alatt a jobb oldalon 'KEMPSTON' felirat látható, vagyis alaphelyzetben erre az Interface-re definiálták a játék vezérlését. Ha megnyomjuk a 'J' billentyűt, megjelenik az 'INTERFACE 2' felirat, újabb 'J' megnyomására pedig a ' CURSOR KEYS' szöveget olvashatjuk. Ha nem rendelkezünk Interface-szel, értelemszerűen Interface 2 esetén a 6/7/8/9/0 (bal/jobb/le/fel/tűz), míg a kurzor billentyűk esetén az 5/6/7/8/0 (bal/le/fel/jobb/tűz) vezérlések adódnak. Ilyenkor KEPMSTON-t ne válasszunk, mert megkergül a játék. Ha megnyomjuk az 'L' billentyűt, akkor a nehézségi szintet állíthatjuk be, a 'LEVEL' felirat mellett megjelenik az 1/2/3 számok valamelyike. '1' esetén könnyű, '2' esetén közepes, míg '3' esetén nehéz szintet kérhetünk.



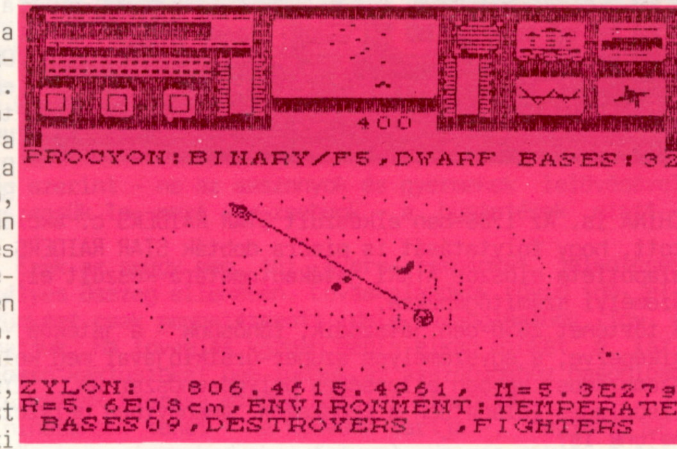
Vegyük szemügyre egy kicsit a műszerfalat:

A bal felső sarokban az űrsiklónk energiaszintjét (POWER) tekinthetjük meg. Ha a vízszintes sáv csökken, az energiánk fogy. Alatta az atombombáink számát kijelző műszert helyezték el, összesen 16 pár bombával rendelkezünk, ezért használatukat csak megfontolt esetben javasoljuk. Ettől jobbra, és a radar másik oldalán is egy-egy fegyver-hőmérséklet jelzőt látunk. Kettő van belőle, hiszen a lézerágyúnk ikercsöves. Nem szabad, hogy az ágyúk túlmelegedjenek,

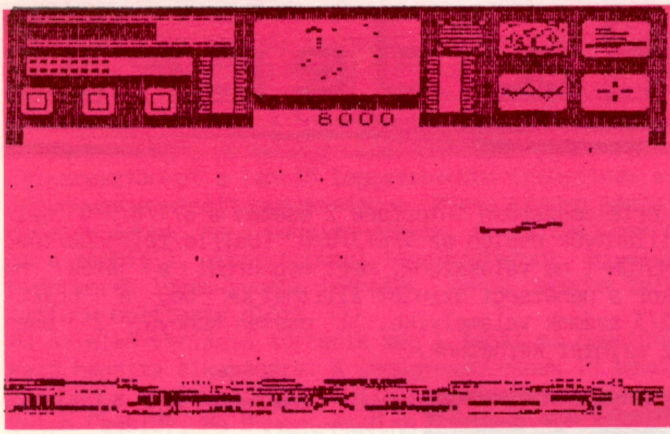
2.2 STAR RAIDERS II.

mert akkor nem tudunk velük lőni. A hőmérők között elhelyezett kombinált műszernek kettős feladata van. Alaphelyzetben - és a játék bejelentkezésekor is - radarernyő, melyen magunk és ellenségeink pozícióját tájolhatjuk be. Ha viszont megnyomjuk a 'T' billentyűt, egy un. sérülésjelző műszert kapunk, amelyen az űrsikló képe jelenik meg. Sérülés esetén a sérült területek, vagy az energiapajzs sérülése villogó pontok formájában tájékoztat minket a sérülés minőségéről és nagyságáról. A 'T' billentyű ismételt megnyomására visszatérhetünk a radarernyőre. A kombinált műszer alatt a mindenkori pontszámunk látható.

A pontszám alatt jelennek meg a gép üzenetei, pl. 'kritikus energiaszint', stb. A küldetést a 'CAPS SHIFT' billentyűvel tudjuk indítani, ez a billentyű játék közben a játékból való kiszállást is jelenti. Ha megnyomjuk a 'SPACE' billentyűt, a képernyőn megjelenik a csillagrendszer térképe. Ezen a térképen középen látható a nap, körülötte ellipszis alakú pályán bolygók keringenek. A keringés vonala a térképen pontokkal jelölve van. A hipertérben igen hasznos közlekedési mód az un. hiper-ugrás, így hamar eljuthatunk bármilyen távoli bolygóhoz, vagy más objektumhoz. Az ugrást úgy tudjuk végrehajtani, hogy ki



kell jelölnünk a cél-objektumot (ez a bal/jobb/fel/le funkciókkal lehetséges). A cél-objektumhoz vezető elképzelt útvonal a térképen meg is jelenik pontozott vonal formájában. A cél-objektumról néhány információ jelenik meg a képernyőn, majd a tűzgomb segítségével hajthatjuk végre a hiper-ugrást. A középen látható bolygó a nap. Ha lehet ide ne szándékozzunk ugrani, mert a gép közli velünk, hogy a hőmérséklet túl magas, és elolvadtunk. A kisebb méretű színes körök a bolygókat jelölik. Ha célunk egy ilyen bolygó, a gép néhány információt közöl a bolygóról, köztük megtudhatjuk az ellenséges gépek számát is. Amennyiben van a bolygón ellenség, tisztítsuk meg tőlük a bolygót, mert szétlövik a városainkat és ismét vége lesz a játéknak.



A négy, egymás mellett villogó pont a ZYLON-ok jelenlétére figyelmeztet. Ha megnézzük őket közelebbről, találkozunk vadászszal: mozgékony, de mivel csak lézerrel lő, ezért nem túl veszélyes. Az un. 'robbantó' egy repülő csészealjhoz hasonlít. Ez erősebb, néhány kellemetlen percet okozhat nekünk. Ellene ionágyúval harcolhatunk - a gép automatikusan átváltja a fegyvert, ha szükség van rá. A ZYLON főhajó nagyon erős, célszerű gyorsan lelőni. A térképen kis méretű, fehér téglalapot is láthatunk, ez az űrállomás. Itt javíthatjuk meg sérült gépünket, bármikor, ha leszállunk. Vigyázzunk, mert ha az ellenséges gépeket túl közel engedjük az űrállomáshoz, akkor szétlövik azt. A különálló kis világító pontok a szomszédos csillagrendszert jelzik. Ide át kell ugrani, s ebben a rendszerben további ugrásokat végezhetünk. Célunk a bázisok megsemmisítése. Ehhez szükségünk lesz a védőpajzsra, melyet az 'S' billentyűvel kapcsolhatunk be/ki, ill. az atombomba üzemmódra is, mely a 'W' billentyűvel kapcsolható be/ki. Atombomba üzemmódban a tűz-gomb az atombombát ejti le.

Ajánljuk a játékot mindenkinek, de elsősorban botkormány tulajdonosoknak!

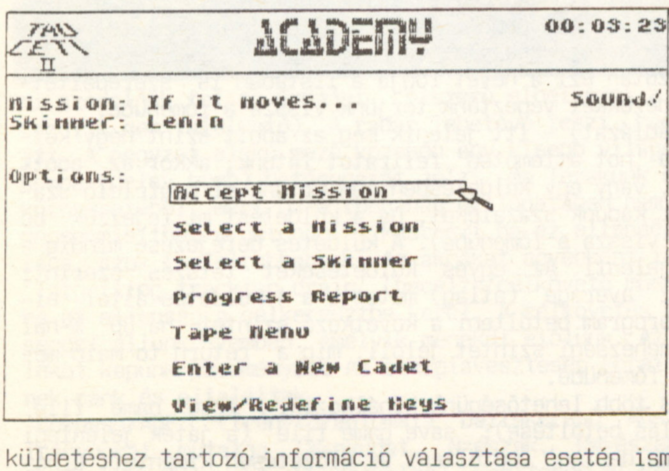
2.3 ACADEMY



A 'CRL' (Computer Rentals Ltd.) csoport első sikerét 1983-ban a DERBY DAY c. BASIC programjával alkotta. Azóta több igen színvonalas játék került ki a kezük közül. Ezek közé sorolhatjuk a TAU CETI sorozat tagjait is, melynek 2. része ACADEMY címmel 1986. végén látott napvilágot. Mielőtt megismerkednénk az ACADEMY-vel, előbb tekintsünk vissza elődjére, az egy évvel korábban megjelent TAU CETI-re: 2050-ben a gyarmatosítók egész raja hagyta el a földet, hogy a szomszédos TAU CETI naprendszerben egy új világot keressen. A TAU CETI 111 első látásra barátságtalan, sivatagos bolygó lehetőséget biztosított a hódításra. A gyarmatosítók 19 év alatt 30 várost építettek fel, ám 19 év múltán járvány terjedt el közöttük, amely tekintélyesen megtizedelte a népességet. A kétségbeesett túlélőket evakuálták. A bolygó-tanács megoldást keresett, azt viszont helyben nem találtak, így elhatározták, hogy expedíciót indítanak a földre. Az expedíció el is indult, de a bolygó - közben meghibásodott - védelmi rendszere elpusztította az expedíciós űrhajót. A bolygó-védelem vezérlő rendszere egy masszív atomreaktor komplexumban lett elhelyezve, melyet szintén a meghibásodott rendszer őriz. Feladatunk egy önkéntes pilóta személyében, különféle fegyverekkel felszerelt űrsiklóval bejutni a reaktorhoz, és hatástalanítani a védelmi rendszert.

Nos 163 év közben eltelt, a bolygó megmenekült, sőt több szomszédos rendszerbeli bolygón is megjelentek a gyarmatosítók. 2213-ban sajnálatos esemény történt, egy tapasztalatlan pilóta rosszul ítélte meg sebességét, amikor a Cygnus 61 bolygón landolást végzett a reaktor közelében, s ennek következtében felbolygatta a reaktor szabályzó rendszerét, ami a fél bolygót tüzes lávává változtatta. A bolygóközi-tanács összedugta a fejét, és arra a megállapításra jutottak, hogy létrehoznak egy akadémiát, ahol magas szinten képeznek majd ki pilótákat különleges feladatokra, hogy az előbb említett, vagy hasonló eset máshol ne fordulhasson elő.

A játékban egy ilyen kadét szerepét töltjük be, 5 nehézségi szinten összesen 20 küldetést kell teljesítenünk. Ha mind a 20 küldetést sikeresen teljesítjük, az akadémia végzett növendékei közé kerülünk, és bevethetők vagyunk különleges feladatok végrehajtására.



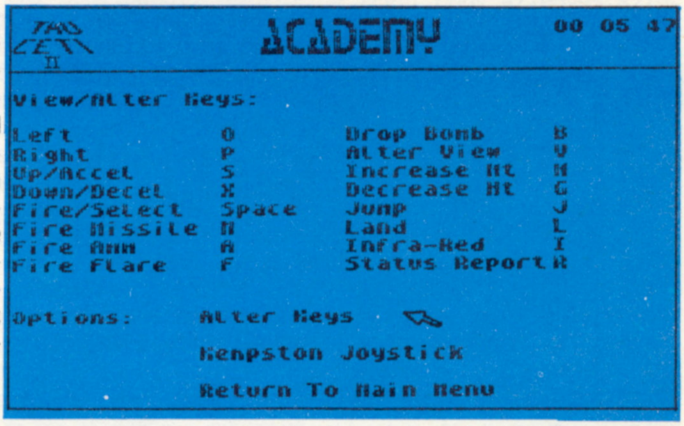
Betöltés után megjelenik a főmenü, melyben 7 opció közül választhatunk. ' Accept mission ' (elfogadni a küldetést): ezzel indítjuk a játékot. Felette a küldetés megnevezése és az űrhajó neve látható.'Select a mission' (Kiválasztani egy küldetést): az adott nehézségi szinten belül 4 küldetés közül választhatunk. Alul további 3 választásunk adódik: -Info on this mission (Információ a küldetésről); -Load in next level (A következő nehézségi szint betöltése); -Selection complete (A választás befejezése). Az adott küldetéshez tartozó információ választása esetén ismét három lehetőségünk kínálkozik: 'Text' - szöveges információ kérhető a küldetés körülményeiről; 'System' - Információ kérhető a rendszerről, amelyben a küldetést végre kell hajtanunk.

2.3 ACADEMY

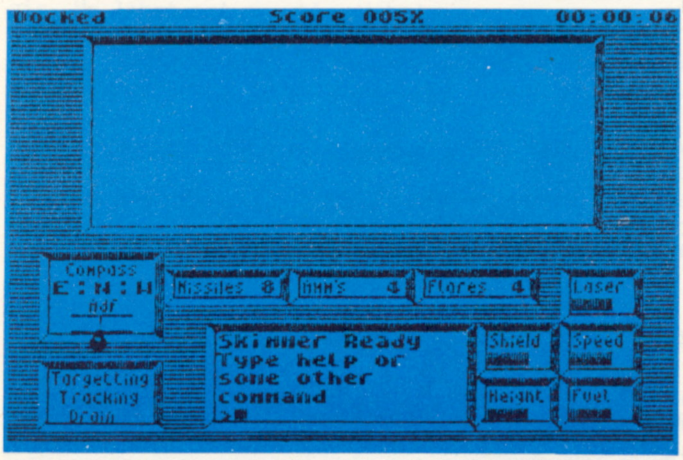
'Select a skimmer' (Kiválasztani egy űrhajót): megjelenik három űrhajó név, és három nem használt (unused) felirat. A három űrhajó (Lenin, Lincoln, Wilson) azok közé az űrhajók közé tartozik, amit az akadémia adott nekünk. Ezeket átépíteni nem lehet. Amikor információt kérünk a küldetésről, akkor a számítógép e három űrhajó közül ajánl egyet. A menü a továbbiakban még négy lehetőséget kínál: 'Info on this skimmer' (Információt kérni az űrhajóról); 'View panel' (Megtekinteni a műszerfalat); 'Design new skimmer' (Új űrhajó tervezése); 'Section complete' (Visszatérés a főmenübe); Ha információt kérünk az űrhajóról, táblázat alakjában megjelenik, mit is tartalmaz az űrhajó. Itt a megnevezések mellett a no/yes (nem/igaz) felirat, illetve a low/middle/high vagy none/4/8-as szerepel. A no/yes megmondja, hogy az adott műszerrel fel van-e szerelve az űrhajó. A low/middle/high (alacsony/közepes/magas) a műszerek teljesítményére utal, ill. a none/4/8 (nincs/4/8) a fegyverzetek lőszereit mutatja meg. A megjelenő lista sorban: scanner (radar); compass/ADF (iránytű/automata bázis-kereső); target/track unit (becélzás/célkövetés); jump/door unit (ugrás/ajtó egység); infra red unit (infravörös kamera). A fegyverek: missiles (rakéták); AMM's (automata rakéta elhárítók); flares (világító rakéták); delay bombs (késleltetett bombák). Az alapműszerek: laser power level (lézerenergia-nagyság); low (1 lézerágyú van), middle (2 lézerágyú van), high (3 lézerágyú áll rendelkezésre). Main drive power (gyorsaság); shield power unit (pajzs ellenálló képesség); steering unit (kormányzás gyorsasága); Return to menu-re az előző menübe lépünk vissza.
'View panel' (megnézni a műszerfalat) - a műszerfal felépítéséről tájékoztat. 'Design new skimmer' (Új űrhajó tervezése) - itt az a lista jelenik meg, mint amikor információt kérünk egy űrhajóról, viszont ez a lista szerkeszthető. A listában egy vonalban, jobb oldalon látható az egyes műszerek nehézsége (weight). Az össz nehézségnek nem szabad a 100 egységet meghaladnia (ez a 'total weight' felirat mellett ellenőrizhető). Minden műszernek más az ára, fontosság szerint, és az árnak sem szabad a 100 egységen túlhaladnia. Miután felszereltük a műszerekkel az űrhajót (természetesen a nyíl segítségével) és a legjobban kihasználtuk az űrhajó teherbíró képességét, ill. a pénztárcánk által nyújtott lehetőségeket, a 'design complete' (befejezni a tervezést) segítségével tovább léphetünk. Ha közben meggondoltuk magunkat, az 'abandon design' (abbahagyni a tervezést) segítségével visszaléphetünk. Ha továbblépünk, 3 műszerfal mintázat (panel texture) és 8 szín közül (panel colour) választhatunk. A 'place instruments' (elhelyezni az eszközöket) lehetőséget biztosít arra, hogy a műszerfalon tetszés szerint helyezzük el műszereinket. Eközben természetesen az 'abandon design' segítségével ismét visszaléphetünk. Ha mégis folytatjuk a tervezést, megjelenik a csupasz műszerfal, műszerek nélkül. Egy kis ablakban jelenik meg az adott műszer neve, alatta pedig az 'undo last step' (utolsó lépés törlése) felirat. Az első műszer az ablak. Ezt célszerű valamelyik sarokba eltenni, helynyerési szempontból. Szimpatikus lehet a bal-felső sarok. Az ablak (view screen) elhelyezése után az üzenő ablak (message window) helyét választhatjuk ki. Ezek után az előre kiválasztott műszerek helyezhetők el sorban. A végén a 'design complete' (tervezés befejezése) opcióval kilépünk a tervezési üzemmódból. Ha ez is megvolt, a program bekéri az űrhajó nevét. Ezután ezt a nevet fogja a listában is szerepeltetni, a saját űrhajói alatt. Ha mindezekkel végeztünk térjünk vissza a főmenübe.
'Progress report' (teljesítmény táblázat) - Itt jelenik meg az adott szint négy küldetésének aktuális álláspontja. Ha 'not attempted' feliratot látunk, akkor az adott küldetéssel még nem próbálkoztunk, vagy egy küldetésben nem értünk el megfelelő százalékot. Egy küldetésre csak akkor kapunk százalékot, ha a küldetést mi fejezzük be (tehát nem kilövés által kerülünk vissza a főmenübe). A küldetés befejezése mindig a dokkoló bázisban való leszállást jelenti. Az egyes küldetéseket tetszés szerinti sorrendben teljesíthetjük. Alul az 'average' (átlag) mutatja a 4 küldetés által elért átlagot. Csak akkor engedi a program betölteni a következő szintet, ha 80 %-nál feljebb kerülünk. A 'Level...' a nehézségi szintet jelöli, míg a 'return to main menu' segítségével visszakerülünk a főmenübe.
'Tape menu' (szalag menü) - Itt is több lehetőségünk kínálkozik: 'load game file' (egy előzetesen kimentett játékállás betöltése); 'save game file' (a játék jelenlegi pozíciójának kimentése); 'load ship design' (betölteni az előzetesen kimentett űrhajókat); 'save ship design' (kimenteni a tervezett űrhajókat); 'return to main menu' (visszatérés a főmenübe).

2.3 ACADEMY

'Enter a new cadet' (Új kadét benevezése) - A beiratkozás időpontja (date application): 7/11/2047, majd bekéri a nevet (name); ha ezt is beírtuk, bekéri a program a születési dátumot (date of birth) nap/hónap/év formában. Ezek után kiírja, hogy a születés dátuma milyen napra esett. A végén megjelenik a 'Restart from Level I. (Y/N)' (I.szinten kezdjük? Y/N) kérdés. Választásunk után a gép kiírja a 'cadet accepted' (kadét elfogadva) üzenetet, és visszatérhetünk a főmenühöz.



'View/Redefine keys' (Billentyű-vezérlések megtekintése és átdefiniálása); Ha a nyíllal az 'Alter keys' opciót választjuk, átdefiniálhatjuk a beállított billentyű-vezérléseket. A KEMPSTON joystick interface is kiválasztható, illetve visszatérhetünk a főmenübe. Ha a billentyűket átdefiniáltuk, CAPS SHIFT+BREAK-re visszakapjuk az eredeti beállításokat. A funkciók sorban a következők: left (balra), right (jobbra), Up/Accel (fel/gyorsítás), Down/Decel (le/lassítás), Fire/Select (lövés/kiválasztás), Fire Missile (rakéta kilövése), Fire Amm (rakétaelhárító kilövése), Fire Flare (világítórakéta kilövése), Drop Bomb (bomba leejtése), Alter View (nézőpont változtatása), Increase Ht (magasság növelése), Decrease Ht (magasság csökkentése), Jump (ugrás), Land (leszállás), Infra-red (Infravörös kamera), Status report (helyzetjelentés);



Mielőtt elkezdenénk a játékot, tekintsük át a műszerfal felépítését:
Az ábra a 'LENIN' űrhajó műszerfalának felépítését mutatja, melyen észrevehetjük, hogy több olyan műszert nem tartalmaz, mely az előbb szerepelt a felsorolásban. A műszerfal felett bal oldalon láthatjuk a helyzeti üzenetet. Felszállás előtt 'Docked' feliratot látunk itt. Középen az eddig elért százalékunk látható, míg jobb oldalon az idő múlását kísérhetjük figyelemmel. A legnagyobb területű téglalap alakú mező az ablak (view screen). Ha mi szereljük fel az űrhajót, célszerű elhelyezni a radart (scanner) a műszerfalon (ez itt nem található). A radar lehetővé teszi, hogy láthassuk az ellenség pozícióját. A négyzet alakú mező közepén egy kisebb világoskék színű mező a késleltetett bomba (delay bomb) hatósugarát jelzi. Ha lerakunk egy ilyen bombát, siessünk el a környezetéből, mert ha a robbanás pillanatában nem vagyunk elég távol, minket is megsemmisíthet a detonáció. Másrészt ha az ellenséges aknák belekerülnek a hatósugarába, azok is felrobbannak. Az aknákat egyébként is illik nagy ívben kikerülni.
'Targetting tracking drain' (becélzó/célkövető műszer); A célzás (targetting) jelez, ha az ellenség a célkeresztbe kerül. A célkövetés (tracking) jelez, ha olyan ellenséggel állunk szemben, amelyik menekül előlünk. Az ilyen ellenséget szétlőve százalékot kapunk eredményül. Az energiaveszteség (drain) akkor jelez, ha valahonnan lőnek ránk és eltaláltak.
'Compass/ADF' (Iránytű/automata báziskereső műszer); Az iránytű a haladási irányunkat mutatja (W=nyugat, N=észak, NW=északnyugat, E=kelet, NE=északkelet, S=dél, SW=délnyugat, SE=délkelet). Segítségével könnyebbé válik az űrhajó tájolása. A bázis hullámokat bocsájt ki, amit az ADF (automata báziskereső) műszer fog fel. Az

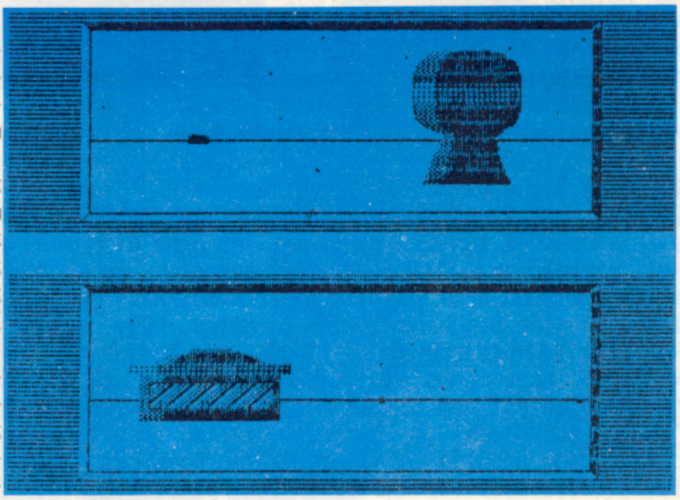
ADF műszeren egy kis vonal jelzi a bázis felé vezető irányt. Akkor haladunk egyenesen a bázis felé, ha ez a vonal középen van.
'Jump pad/Door unit' (Ugrás/ajtó műszer); Az ugrás (jump) akkor jelez, ha űrugrást hajtunk végre. Figyelem! Ugrani csak a műszer felszerelése után lehet, így pl. a LENIN űrhajó esetében a próbálkozás hatástalan. Az ajtó (door) műszer akkor jelez, ha túl gyorsan közeledünk a bázis felé. Vigyázzunk, mert ha ebben az esetben nem lassítunk, a küldetésnek vége.
'Missiles' (rakéták); A műszer számlálja, hogy hány rakétánk maradt még felhasználatlanul. A rakéta a célkeresztben lévő ellenséget támadja és semmisíti meg. Ha egyszer kilőttük, automatikusan a céltárgyra ragad. Ezután elmozdulhatunk a célkereszttel, de az ellenség is mozoghat, a találat biztos. Az ellenség persze védekezhet is, a lézer hatástalanítja kilőtt rakétáinkat. Ha a túlzott lézerzápor miatt megsérülnek - még az űrhajón lévő - rakétáink, úgy a műszer zérus értékre vált.
'AMM's' (rakéta elhárítók); A műszeren a kilőhető rakétaelhárítók számát látjuk. Ellenséges rakétákat semmisíthet meg. Láthatatlan fegyver, ami olyan impulzust küld ki az űrhajóból, ami az összes rakétának beindítja az önmegsemmisítő rendszerét. Ha az AMM's meghibásodik, a műszeren zérus értékünk lesz.
'Flares' (világító rakéták); ld. 'Infravörös kamera'
'Delay bomb' (késleltetett bomba); A késleltetett bomba azon a ponton marad, ahol leejtettük, és egy bizonyos idő elteltével felrobban. Célszerű ilyenkor a 'drive power'-t (sebesség) a legmagasabb szintre tenni. A késleltetett bomba csak egyetlen ellenségre hatástalan, de ez az ellenség nem is támad, lehetetlen megsemmisíteni, kizárólag akadályként lett elhelyezve az űrhajó számára.
'Laser' (lézer melegedésmérő); A lézerfegyver felmelegedését méri. Ha elérte a végső határt, a fedélzeti számítógép nem engedi, hogy addig lőjünk vele, amíg egy kicsit le nem hűlt.
'Fuel' (energia); A még rendelkezésre álló energiánk szintjét mutatja. Energiát a bázison tudunk tölteni.
'Speed' (sebesség); Az űrhajó mindenkori sebességének kijelzésére szolgál.
'Height' (magasság); Az űrhajó mindenkori - földtől számított - magasságának kijelzésére szolgál. Leszálláskor ügyelni kell, ne túl nagy sebességgel dobjuk az űrhajót a földre, mert meghibásodhat.
'Shield' (védőpajzs); Az űrhajó védőpajzsának épségét jelzi. Ha a mező csökken, nagy az esélyünk arra, hogy előbb-utóbb kilőnek.
'Infra red system' (Infravörös kamera); Jelez, ha bekapcsoljuk. Ez is csak akkor működik, ha rajta szerepel a műszerfalon. Általában este van rá szükség. Ilyenkor minden egyszínű, árnyéktalan lesz. A forgás zavarja az infravörös kamerát, elcsúszásokat eredményezhet, amit persze a fedélzeti számítógép korrigál. A világító rakéta jobb, mint az infravörös kamera, de rövid ideig tart a hatása. Annyiban mindenképpen jobb, hogy nem eredményez elcsúszásokat, de mégis az infravörös kamera használatát ajánljuk, mivel csak korlátozott számú világító rakéta áll rendelkezésünkre.
'Message window' (üzenő és input ablak); Itt tudunk kommunikálni a fedélzeti számítógéppel, ill. itt tájékoztat minket a fedélzeti számítógép az általa fontosnak tartott információkról (pl. hogy az ellenség rakétákat lőtt ki). Az információ gyakran hasznos lehet, mert pl. nem ér váratlanul minket egy esetleges hátulról jövő támadás. A játék indításakor az ablak input módban van, a 'HELP' szó beírásával (+ENTER) kiírathatjuk a számítógép számára alkalmazható parancsokat. A SIGHT ON-t célszerű rögtön kiadni, mert ez a célkeresztet kapcsolja be, s ez igen hasznos. Tekintsük most át a megadható parancsokat:
HELP (segítség): kiírja a megadható parancsok listáját; LAUNCH (kilövés): elindítja az űrhajót a bázisról; PAUSE (szünet): megállítja a játékot; QUIT (kilépés): kiszáll a játékból; SIGHTS ON (célkereszt be)/SIGHTS OFF (célkereszt ki): fontos, a célkereszt mindig legyen bekapcsolva; WAIT (várni): ez akkor fontos, ha elfelejtettünk infravörös kamerát felszerelni, vagy az űrhajó nem rendelkezik ezzel az eszközzel, elfogyott a jelzőtüzünk, és lement a nap is. Ekkor egy biztonságos helyen álljunk meg, ereszkedjünk, majd szálljunk le, és várjuk meg, amíg újra feljön a nap. Ekkor folytathatjuk a küldetést. EQUIP (felszerelés): ezzel a lehetőséggel - csak a bázisokban - felszerelhetjük az űrhajót. A 4 fegyver mellett REPAIR-t (javítás) és REFUEL-t (üzemanyag feltöltés) kérhetünk. STATUS (helyzet): helyzetjelentést kérhetünk az űrhajó állapotáról és a környezetről. CODES (kód összerakása): erre a II.szinten

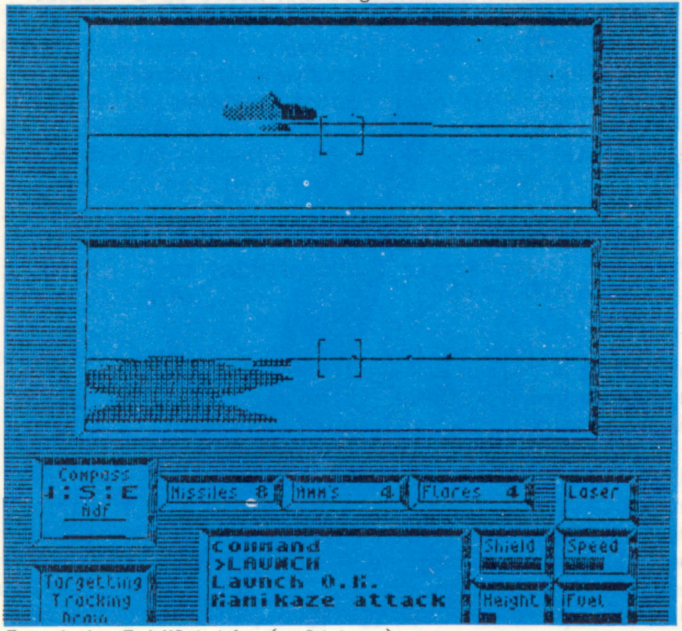
lesz szükség. DEAF: ez a parancs akkor lép életbe (automatikusan), ha reaktorokban landolunk, illetve oda szállunk be kódokat összegyűjteni. A meghatározott dallamokat visszajátszva megtörik a kódzár, belépést nyerhetünk és megkapjuk a kódot. LOOK (nézni): jelentést kapunk a helyzetünkről, pl. az űrhajó lelandolt a bolygón.

LAUNCH (+ENTER) indítja a beállított küldetést. Sok levélíró javaslata alapján most nem a teljes megoldás leközlésésére törekszünk. Úgy döntöttünk, hogy az I-II és az V. szint megoldásához nyújtunk némi segítséget. Tekintetbe véve, hogy az egyes szintek egymásra épülnek, úgy érezzük, aki eljut a II.szintig, az nagy valószínűséggel be is tudja fejezni a játékot.

I.szint, 1.küldetés (If it...)

Célunk kiirtani minden ellenséget, amit a pályán találunk. Általános robot támadásra számíthatunk. Javaslatunk, hogy lassan közelítsük meg az ellenségeket, mert így könnyebben végezhetünk velük. Az ellenségek több helyen tömörülnek, itt csak egyenként tudjuk kiirtani őket. Több robot kiirtása után gyakran jelenik meg az üzenő ablakon a 'KAMIKAZE ATTACK' felirat, ami azt jelzi, hogy olyan űrhajókkal van dolgunk, amelyek szívesen feláldozzák magukat azzal, hogy belénk jönnek. Még egy jó tanács. A képen látható lapos építmény a bázis. Ha lehet ezt ne támadjuk !



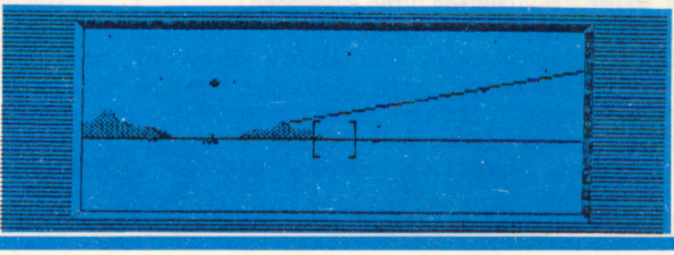


I.szint, 2.küldetés (red dawn)

Célunk megkeresni, és elpusztítani a gyárakat. Gyárakat északra, nyugatra, északnyugatra és keletre találhatunk. Ezeket jól őrzik. Gyárak elpusztításához először is ki kell irtanunk minden ellenséget a gyárak körül. Ha ez megtörtént, álljunk meg a gyárral szemben, és ha várakozunk egy kicsit, a pajzs felmegy. Ekkor teljes sebességgel induljunk el a gyár felé. Közel érve (ha két gyár van egymás mellett, akkor a két gyár közé érve) dobjuk le a bombát. Végül távolodva, a view-val hátratekinthetünk, és még szétlőhetjük a hátramaradt rakétáinkat is. Az ugráshoz használjuk a JUMP funkciót, de ehhez a műszerrel is (jump/door unit) rendelkeznünk kell.

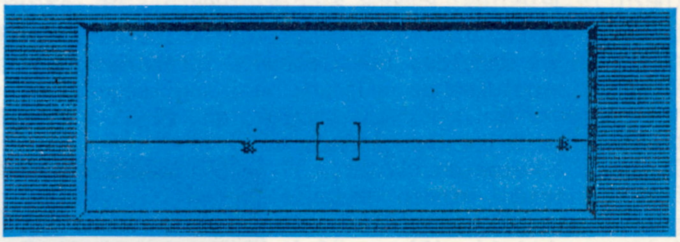
I.szint, 3.küldetés (meltdown)

Célunk eljutni a reaktorhoz és megsemmisíteni azt, mindössze 15 perc alatt, mert különben a reaktor felrobban. A reaktort két módon semmisíthetjük meg: 1.bombát helyezünk el mellette és elrepülünk; 2.Igen közel megyünk hozzá és az utolsó pillanatban



egy rakétát lövünk ki rá. (Ez utóbbi azért szükséges, mert a reaktor rakétaelhárítóval védekezik, ez pedig hatástalan, ha nagyon közel vagyunk hozzá.

I.szint, 4.küldetés (softly...)
Célunk eljutni a bázishoz egy elaknásított mezőn keresztül. Az úton elszórva kisebb-nagyobb ellenállásokba ütközhetünk. Az aknák hatástalanítására igen hasznos lehet a késleltetett bombák használata. A bázis tájolását az ADF műszer segítségével végezhetjük el.



Minden küldetést csak úgy fejezhetünk be, ha visszaszállunk a bázisra.

II.szint, 1.küldetés (cipher)
Célunk, hogy 12 kódot összeszedjünk, majd ebből 4 számkódot összerakjunk. A kódokat a reaktorokból tudjuk begyűjteni. Az űrhajót ehhez a művelethez mindenképpen fel kell szerelnünk a 'jump/door' műszerrel, a reaktorokhoz való bejutáshoz. Egy-egy reaktor esetében a bejutáshoz zenei kódot kell megfejtenünk. A reaktorokat jól védik igy a bejutás nem könnyű dolog. Fontos, hogy ne lőjünk bele a reaktorokba, mert akkor az ajtók bezáródnak, s a kódot nem tudjuk összegyűjteni. Ennek az lesz a következménye, hogy nincs együtt a 12 kód, azaz nem tudjuk összerakni a számkódokat, tehát a küldetést sem tudjuk befejezni. Ha összegyűjtöttük az összes kódot, akkor a 'CODES' paranccsal (amit csak leszállás után lehet beírni) lehet összeállítani a kódokat. Jobb oldalon láthatjuk milyen kódokkal rendelkezünk, és ezeket a felfelé/lefelé vezérléssel váltogathatjuk. A 'place' segítségével a kód bal oldalra helyezhető. A kód mellé illő többi kód színét az alapkódhoz egyeztetnünk kell, ez a 'colour' paranccsal történik. Ha a színek is egyeznek, akkor a szerintünk odaillő kódnak megfelelően kell elforgatnunk a bal oldalon lévő kódot. Ha egyeznek, a 'place' segítségével ismét rátehetjük az első kódra a másodikat. Tehát a könnyebb érthetőség kedvéért: először a két kód színe egyezzen, utána a bal oldali kódot forgassuk el a megfelelő pozíciójába és a 'place'-szel helyezzük el a második kódot is. A számkód egy részét három kód segítségével tudjuk előállítani.

.
.
.

V.szint, 1.küldetés (the coal mine)
Célunk, hogy megsemmisítsük az ellenséges robotokat.

V.szint, 2.küldetés (pass)
Célunk az ellenséges erők megsemmisítése. A pályán sok ugráló robot található. Vigyázat! A IV. szinttől az ellenség már szuper-rakétákkal is rendelkezik, ami ellen nincs védekezés! Ha szuper-rakétákkal tüzelnek ránk, célszerű elmenekülni.

V.szint, 3.küldetés (protector)
Célunk 8 db. 'tracker' szétlövése, ezután pedig 'élve' vissza kell jutnunk a bázisra. A 'tracker'-t úgy tudjuk észrevenni, hogy figyeljük a radar-ernyőt. Ha valami gyorsan távolodik tőlünk, üldözőbe kell venni, és meg kell semmisíteni. a 8 db. 'tracker' kb. egy vonalban egyenesen előre található.

V.szint, 4.küldetés (the sheapard)
Célunk, hogy 5 megfigyelőtornyot visszatereljünk a bázis közelébe. Célszerű kezdetben az ellenségektől kiüríteni a környéket, majd hatástalanítsuk az aknákat is. Ezt úgy hajtsuk végre, hogy egyszerűen menjünk neki az aknának, de egyszerre csak egynek. Az akna felrobban, a pajzsunk 50 %-a ezt megsínyli, de ha várakozunk egy kicsit, a pajzs ismét feltöltődik, és próbálkozhatunk a következő aknával. Ha megtisztítottuk a terepet, a tornyokat üldözzük vissza lézer segítségével a bázishoz.

Ha a feladatot teljesítettünk, átmentünk a vizsgán. Sok sikert a tanfolyamhoz !

-néma sprite: olyan sprite, amely nem tartalmaz a képernyő számára értelmezhető adatokat. Gépi kódú rutinokban, tömbökben, vagy mint ütközéseket vizsgáló sprite-ként használhatóak.

A SPRITE-OK ELHELYEZKEDÉSE A MEMÓRIÁBAN

Mielőtt részletesebben beszélnénk az IDEAL parancsairól, vizsgáljuk meg, milyen formában kezeli a rendszer a sprite-okat:

A WHITE LIGHTNING sprite-jai a memória egy különálló területén, a sprite-memóriában helyezkednek el, aminek a kezdőcímét az SPST, a végcímét az SPND határozza meg. A sprite-memória leginkább egy nagy fiókos szekrényre hasonlít: minden fiók egy-egy sprite adatait tartalmazza, amelyekre a fiókhoz rendelt számmal (SPN) lehet hivatkozni.

Az egyes sprite-ok a következő adatokat tartalmazzák:

- az első byte a sprite száma (SPN). A sprite-oknak nem fontos szám szerinti sorrendben elhelyezkedniük, tehát pl. a 85-ös számú sprite-ot követheti a 4-es is. A rendszer összesen 255 sprite-ot tud kezelni, amelyekre az 1-255 számokkal hivatkozhatunk
- a második és harmadik byte tartalmazza a következő sprite kezdőcímét
- a negyedik byte határozza meg a sprite szélességét (LEN)
- az ötödik byte tartalmazza a sprite magasságát (HGT)

Ezután található a sprite képpont adatait tartalmazó adatblokk. Az adatok tárolása karakterenként történik, tehát könnyen belátható, hogy a képpontadatok 8 byte (1 karakter) x szélesség x magasság hosszon helyezkednek el.
A képpontok után szélesség x magasság hosszon helyezkednek el a sprite attributumai.
A fentiekből kitűnik, hogy egy sprite adatainak a hossza a sprite-memóriában:

5+(9xszélességxmagasság)

Ha függőleges pixel- vagy attributum-scrollozást hajtunk végre, az eltűnő adatokat a későbbi visszatéréshez a programnak el kell tárolnia. Erre szolgál a puffer, ami a sprite-memória felett helyezkedik el, kezdőcímére az SPND mutat. A puffer méretét (alapértelmezésben 256 byte) a program elején beállíthatjuk úgy, hogy végre tudjuk hajtani a használni kívánt legnagyobb scrollt is.

Nézzünk egy példát a scrollozásra használt pufferméret kiszámítására: a scrollozott attributumok oszloponként egy byte-ot használnak, amelyeket meg kell szoroznunk a pixelscroll nagyságával (az NPX-ben beállított értékkel, az előjel nem számít). Tehát ha egy 4 karakter széles sprite-ot vagy képernyő-ablakot scrolloztatunk 10 pixellel, a szükséges memóriaterület 4 x 10 = 40 byte.

AZ 'IDEAL' VÁLTOZÓK

Az IDEAL programnyelv a sprite- és grafikai műveletek támogatására több törzsváltozót használ:

ROW: egy ablak bal felső sarkának függőleges (y) koordinátáját határozza meg (0-23)

COL: egy ablak bal felső sarkának vízszintes (x) koordinátáját határozza meg (0-31)

LEN: az aktuális ablak szélességét tartalmazza (1-32)

HGT: az aktuális ablak magasságát tartalmazza (1-24)

SPN: az aktuális sprite száma (1-255)

SP 1: sprite-ok közötti művelet esetén annak a sprite-nak a számát(1-255) határozza meg, amelyet a művelet "céloz"

SP 2: sprite-ok közötti művelet esetén annak a sprite-nak a számát tartalmazza, ahol a művelet eredménye előáll

SROW: a sprite-ablak bal felső sarkának a sprite bal felső sarkához viszonyított függőleges pozíciója

SCOL: a sprite-ablak bal felső sarkának a sprite bal felső sarkához viszonyított vízszintes pozíciója

SLEN: a hidegindítás használatakor a start után törlendő sprite-terület hosszát tartalmazza

MLEN: a relokáció mértékét és irányát tartalmazza. Pozitív szám felfelé, negatív szám lefelé történő relokálást eredményez

SPST: a sprite-terület kezdőcíme

SPND: a sprite-terület utolsó byte-ja után lévő byte címe (egyben az előtér-scroll pufferének a címe)

NPX: a függőleges scroll pixelekben értelmezett nagyságát és irányát tartalmazza (-128 - +127). Pozitív szám felfelé, negatív szám lefelé történő scrollozást eredményez

SPTR: a TEST utasításból visszatérve a sprite kezdőcímére mutat

DPTR: a TEST utasításból visszatérve a pixel adatok kezdetére mutat

SZINTAKTIKA

Az IDEAL nyelv szintaktikája a normál BASIC-ben megszokottól eltérő, fordított rendszerű: a direktíva előtt kell megadni a direktívához tartozó értéket. Az utasításokat felkiáltójellel kell egymástól elválasztani Pl. egy a képernyő bal felső sarkában lévő, 10x10 karakter méretű ablak deklarálása a következőképpen néz ki:

```
0 COL ! 0 ROW ! 10 HGT ! 10 LEN
```

MŰVELETEK A SPRITE-MEMÓRIÁBAN

COLD #

Az utasítás beállítja SPND értékét SPST értékére és törli az SPST feletti memóriát. A törölt memóriaterület hosszát a SLEN tartalmazza.

Tegyük fel, hogy van egy sprite-készletünk, amely 5 db 6x8 karakter méretű sprite-ból áll, amelyekkel 15 pixeles függőleges scroll-okat kívánunk a későbbiekben végrehajtani. Ezt a készletet át kívánjuk helyezni a memória felső részébe. Először ki kell számolnunk az átmozgatandó terület hosszát: mint tudjuk, egy sprite adatai 5 + (9 x szélesség x magasság) hosszon tárolódnak, 5 sprite-unk tehát 5 x (5 + (9 x 6 x 8)) = 2185 byte-ot foglal el a memóriából. A 8 karakter széles sprite 15 pixeles scrolljához 8 x 15= 120 byte-nyi puffer szükséges, összesen tehát 2185 + 120 = 2385 byte-ot kell átmozgatnunk. Ezt az értéket a SLEN-ben kell tárolnunk.

A memória 65520-65535 címei a háttér-rutinok részére vannak lefoglalva, tehát az SPND maximum a 65520 - SLEN címig (esetünkben 63135) terjedhet:

```
63135 SPST ! 3450 SLEN ! COLD # (+ENTER)
```

és az SPND beáll 63135-re

SPRITE

Sprite-ok helybiztosítására használható utasítás, az SPN-ben meghatározott számú sprite-nak biztosítja a HGT és LEN változókban meghatározott memóriahelyet (5 + (9 x HGT x LEN) byte). Nézzünk egy példát az utasítás használatára:

```
48 SPN ! 5 HGT ! 5 LEN ! SPRITE
```

A parancssor helyet készít a 48-as számú, 5 x 5 karakter méretű sprite adatainak. Vigyázat, a végrehajtás után a sprite még nem létezik, még be kell néhány dolgot poke-olnunk. Először ki kell olvasnunk a sprite kezdőcímét, amit az utoljára bevitt sprite 2. és 3. byte-ja tartalmaz ezt elhelyezzük egy "A" változóba, majd beállítjuk az új sprite első 5 byte-ját:

```
LET A = PEEK (utolsó sprite 2. byte-
ja) + 256 * PEEK (utolsó sprite 3.
byte-ja)
```

```
RANDOMIZE (A + (9 * HGT * LEN) + 5)
POKE A+1, PEEK 23670
POKE A+2, PEEK 23671
POKE A+3, (HGT)
POKE A+4, (LEN)
```

A sprite most már néma sprite-ként létezik (a gép elfogadja a rá való hivatkozást), de nekünk kell ezután adatokkal feltöltenünk. Mint látható, a sprite-ok ilyen úton történő bevitele kissé nehézkes, használjuk inkább a sprite-generátort.

ISPRITE

Hasonló a SPRITE utasításhoz, de itt az SPST csökken az új sprite-hoz szükséges memóriacímmel, tehát a sprite-memória lefelé bővül. Ugyanazokat az értékeket kell beállítanunk, mint a SPRITE-nál.

RELOCATE

Egy másik módja a sprite-ok számára történő helybiztosításnak a RELOCATE parancs használata. Ez a MLEN változó által meghatározott érték szerint biztosítja a helyet: ha az érték negatív az SPST mozog, ha pozítiv az SPND. Pl. egy 3 x 3 karakter méretű sprite-hoz (3 x 3 x 9) + 5 = 86 byte szükséges:

```
-149 MLEN ! RELOCATE
```

Megjegyeznénk, hogy a sprite-memóriát növelő utasítások használatánál mindig meg kell győződnünk arról, hogy a bővítéshez elegendő-e a szabad hely, nem nyúlik-e bele valamilyen értékes adatállományba.

WIPE

Megsemmisíti az SPN által meghatározott sprite-ot, lefelé lépteti a felette lévő sprite-okat és frissíti az SPND-t.

DSPRITE

Megsemmisíti az SPN által meghatározott sprite-ot, felfelé lépteti az alatta lévő sprite-okat és frissíti az SPST-t.

TEST

Az egyes sprite-ok vizsgálatát segíti elő a TEST. Az utasítás használatakor a következő paraméterek kapnak értéket:

HGT	magasság
LEN	szélesség
SPST	a sprite-terület startcíme
DPTR	a pixel-adatok kezdőcíme
SIZE	a sprite által elfoglalt memóriaterület

Az igaz-hamis jelzőbit jelzi, hogy a sprite létezik-e vagy sem. Ha a gép nem találja a sprite-ot, a paraméterek értéke nullára áll be.
Az 58. sprite paramétereinek visszahívása pl. a következőképpen történik:

```
58 SPN ! TEST IF HGT ? LEN ? CR ENDIF
```

(A nem IDEAL-parancsok (IF-ENDIF és CR) magyarázatát már megismertük, a Spectraforth-t ismertető részben).

SCROLL MŰVELETEK

Az IDEAL-nyelv segítségével mind vízszintes, mind függőleges scrollokat könnyen végezhetünk. A léptetés nagysága 1, 4 illetve 8 pixel lehet, kigörgetett vagy wrap-scroll. Mivel a scroll-ok ablakokhoz tartozó műveletek, az ablak nagyságát előzetesen minden esetben definiálnunk kell:

COL -az ablak bal szélének oszlopa (0-31)
ROW -az ablak bal szélének sora (0-23)
LEN -az ablak szélessége (1-32)
HGT -az ablak magassága (1-24)

Vizszintes képernyő scroll-ok:

WRLxV - balra scroll x pixellel (wrap)
WRRxV - jobbra scroll x pixellel (wrap)
SCLxV - balra scroll x pixellel (sima)
SCRxV - jobbra scroll x pixellel (sima)

ahol x = 1,2 vagy 4 pixel lehet

Megjegyzés: COL+LEN nem haladhatja meg a 32-t, ill. ROW+HGT nem haladhatja meg a 24-t.

Tetszőleges képernyőtartalom görgetéséhez gépeljük be a következőt:

```
3 COL ! 4 ROW ! 4 HGT ! 4 LEN ! WRL4V
                                 (+ENTER)
```

Függőleges képernyő scroll-ok:

Az előbb említett négy paraméteren túlmenően itt az NPX változóban előzetesen be kell állítanunk a léptetés nagyságát. Ha az érték pozitív, úgy a görgetés felfelé történik, negatív esetben pedig lefelé.

WCRV - függőleges scroll (wrap)
SCRV - vizszintes scroll (sima)

Megjegyzés: Az NPX-nek -128 és +127 közé kell esnie;
Legalább NPXxLEN méretű pufferterületre van szükségünk;
A scrollozási hossz nem lehet nagyobb, mint az ablak mérete;

Attributum scroll-ok:

Végrehajtásuk csak 'wrap' tehát körbegörgetéses módon és csak karakterenként lehetséges.

ATTLV - balra scroll
ATTRV - jobbra scroll
ATTUPV - felfelé scroll
ATTDNV - lefelé scroll

Scroll-ok végrehajtása a memóriában:

Az előbb megismert három scroll-csoport közvetlenül a képernyőn hajtódott végre, erre is utal az utasítások utolsó betűjele ('V'=video). Ha ezt az utolsó betűt 'M'-re ('M'=memory) változtatjuk meg, a megadott scroll a memóriában fog végrehajtódni, méghozzá az aktuális sprite képét fogja befolyásolni.

Előzetesen tehát mindig definiálnunk kell a scroll-oztatni kívánt sprite számát az SPN változóban. Mintapéldánkban görgessük lefelé 11 képponttal az 5. sprite-ot, 'wrap' módban:

```
5 SPN ! -11 NPX ! WCRM
```

Az utasítások szintaktikája egyébként megegyezik az előzőekkel.

MEMÓRIAMOZGATÁSOK

A memóriamanipulációk a képernyő-memória, valamint a sprite-memória közötti adatmozgatást teszik lehetővé különböző módon.
Alapvetően a mozgatás iránya a meghatározó. Ha az utasítás első betűjele 'P' (PUT), akkor az adatmozgatás a sprite memóriából történik a képernyő-memóriába.
Abban az esetben, ha az első betűjel 'G' (GET), úgy a képernyő-memóriából viszünk át adatot a sprite-memóriába.

Memóriamozgatás egy teljes sprite és egy előzetesen definiált ablak között:

PUTBLS- sprite a képernyőre (normál)
PUTORS- sprite a képernyőre (OR)
PUTXRS- sprite a képernyőre (XOR)
PUTNDS- sprite a képernyőre (AND)

GETBLS- ablak a sprite-ba (normál)
GETORS- ablak a sprite-ba (OR)
GETXRS- ablak a sprite-ba (XOR)
GETNDS- ablak a sprite-ba (AND)
ATTON - Memóriamozgatáskor összekapcsolja a képpont adatokat az attributumokkal
ATTOFF- Az ATTON által kijelölt kapcsolatot megszünteti.

A művelet végrehajtása előtt a következő paramétereket kell beállítanunk:

SPN - a felhasználandó sprite száma
COL/ROW- az ablak bal felső sarka
LEN/HGT- az ablak mérete

Memóriamozgatás sprite-ablak és képernyő-ablak között:

PWBLS - spr.ablak a képernyő-ablakba
PWORS - spr.ablak a képernyő-ablakba(OR)
PWXRS - spr.ablak a képernyő-ablakba(XOR)
PWNDS - spr.ablak a képernyő-ablakba(AND)

Ellentétes irányban természetesen a 'P' helyett 'G'-t kell írnunk.

A műveletek végrehajtása előtt az előbb ismertetett változókon túlmenően definiálnunk kell a:

SCOL - sprite ablak bal széle oszlopa
és a
SROW - sprite ablak felső széle sora

változókat is. Itt HGT és LEN a sprite-ablak méreteit fogják meghatározni.

Omnicopy-2

Ez a másoló is rendelkezik több olyan lehetőséggel, amelyek miatt mindenképpen beszélni kell róla. Két részből áll, egy rövid BASIC betöltőből és egy mintegy 3 kbyte hosszú gépi kódú részből. A másoló mechanizmus 2048 byte-ot foglal el a memóriában, ezen túl kb. 600 byte az üzenetek számára lett fenntartva. Az OMNICOPY-2 nem kompatibilis az Interface I-gyel ill. a microdrive-val, így betöltés előtt győződjünk meg róla, hogy az említett perifériák csatlakoztatva vannak-e?
Betöltés után automatikusan indul a másoló. A legfelső sorban üzenetmező, alatta pedig az un. kommunikációs mező látható. Ez utóbbiban jelennek meg a betöltött file-ok adatai. Középen egy táblázatot találunk, melynek bal oldalán az aktuális mód olvasható (betöltéskor 'INPUT MODE'= parancs mód). Ettől jobbra olvashatjuk a betöltött file-ok számát (a másoló külön fileként számítja a 17 byte hosszú fejléceket is!), alatta a dupla sebességű üzemmód bekapcsolt (ON) ill. kikapcsolt (OFF) állapotáról tájékozódhatunk. A táblázat jobb oldalán felül a szabad memória nagysága olvasható (betöltés után értéke 41562 byte). Alatta a bevezető-jel nélküli töltés üzemmód be/kikapcsolt állapota látható (ON/OFF).
A felső üzenetmezőben folyamatosan balra görgetődnek a felhasználható billentyű funkciók, most ezeket tekintjük át:

'L' (LOAD) -Törli a memóriát és áttér betöltési üzemmódba. Betöltés közben folyamatosan megjelennek a fejléc adatok (header data) és a file-ok adatai (file data). Az egyes blokkok (a fejlécek is) azonosító számot kapnak. Betöltés közben mindig csak az adott file adatait látjuk a képernyő kommunikációs mezőjében. Ha megtelik a memória, a program automatikusan visszatér 'INPUT' módba.

'S' (SAVE) - Kimenti az aktuális memóriatartalmat a villogó kurzortól kezdődő filetől az utolsó file-ig folyamatosan. Memóriatakarékos, mivel az egyes file-ok között nagyon rövid szünetet hagy.

'V' (VERIFY) - Összehasonlítja a magnetofonról betöltött file-t a memóriában lévő és kurzorral jelzett file-lal. Hibás eredmény esetén hibaüzenetet jelenít meg.

'C' (COUNT) - Leszámolja a magnetofonról betöltött file hosszát és a képernyő felső részén megjeleníti ezt az értéket. A leszámolt byte-ok nem kerülnek a számítógép memóriájába. Ez tájékozódási információ számunkra, annak megállapítására, hogy egy hosszabb méretű file hosszát megállapítsuk, ugyanis ez az érték befolyásolhatja a megfelelő másoló eljárás kiválasztását. A file hossza egyébként 1-65534 byte lehet.

'B' (BREAK) - Ez egy igen hasznos szolgáltatása a másolónak. Két kérdésre kell válaszolnunk. Az 'ignore:' azt az értéket kéri be, amennyi byte-ot le akarunk vágni egy file elejéből, ill.a 'load:' azt a paramétert várja, amennyi byte-ot innentől kezdve szeretnénk betölteni. Ez lehetővé teszi azt, hogy egy nagyobb file-nak pl. csak a középső harmadát töltsük be ténylegesen.

'K' (KEEP) - Ez az opció gyakorlatilag analóg a 'LOAD' paranccsal, azzal a különbséggel, hogy nem törli a memóriában elhelyezkedő adatainkat. A betöltés a kurzorral jelzett pozíciótól folytatódik.
Vigyázzunk, mert a kurzor utáni adataink itt is elvesznek.

'ARROWS' - A kurzor funkciók (6/7 - le/fel) lehetővé teszik a file-katalógusban való szelektálást.

'Q' (ERASE) - Ez a parancs a kurzorral jelzett file törlésére szolgál. A fejléc file külön törölhető.

'H' (CREATE A HEADER) - A kurzorral jelzett file számára fejlécet generál. A fejlécet gépi kódú programként állítja elő, kezdőcímének 24064-et állít be, hossznak pedig a file hosszának megfelelő értéket. A file-nak a 'Headerfile' nevet adja, és beszúrja a file-ok közé, átsorszámozva a file-ok azonosító számát. A kezdőcímnek nincs sok jelentősége, mert a kimentett file LOAD""CODE 'új cím' utasítással bárhova betölthető.

'R' (DEACTIVATE RUN LINE) - A kurzorral jelzett, és automatikus indítással ellátott BASIC program fejlécéből eltávolítja az automatikus indítás kódját. Ismételt kimentés, ill. visszatöltés után a BASIC listázható.

'D' (DOUBLE SPEED) - A betöltött file-okat 1500 baud sebesség helyett 3000 baud sebességgel menti ki. Ezek a file-ok ezután csak akkor tölthetők vissza, ha a turbó betöltőt megírtuk hozzájuk. Az opció így csak a gyors kezelést teszi lehetővé.

'P' (LEADER PITCH) - Segítségével TONE nélkül is próbálkozhatunk file betöltésével. Gyakori hiba lehet, hogy szalaghiba miatt a betöltés lehetetlen. Ezzel a módszerrel (ON állapot) a program tetszőleges részén beléphetünk a töltés üzemmódba. Az így kimentett file-rész 'rendbehozása' már a programozó dolga.

'N' (MAXFILE) - 41562-49115 byte érték átmásolását teszi lehetővé. Az üzenetek eltűnnek a képernyőről. Betöltés után a program a CAPS SHIFT billentyűvel menthető ki.

'M' (MEGAFILE) - 49116-97615 byte érték átmásolása. Betöltés után a program az 'S' billentyűvel menthető ki, többször is egymás után. Ha 'L'-t nyomunk, újabb hosszú file betöltésére is lehetőségünk van.

'DELETE' - Törli az OMNICOPY-2-t a memóriából.

'SPACE' - Megállítja az üzenetsor görgetését.

'CAPS SHIFT' - Az üzenetsor görgetését felgyorsítja.

Megjegyzés: Az OMNICOPY alkalmas JERKYS (szaggatott TONE-ú) és turbósított file-ok átmásolására is.

SUPER 50K(COPY 86M)



A hibajelző kérdőjel kétszeres megjelenését programozási hiba okozta. A kis hiba az itt közölt BASIC program segítségével kijavítható:

```
10 CLEAR 30000: LOAD "USR63637"CODE: CLS
20 FOR n=64043 TO 64120: POKE n-4,PEEK n: NEXT n: POKE n-3,245
25 POKE n-2,50: POKE n-1,194: POKE n,69
30 GO SUB 60
40 LET n=64036: POKE n,205: POKE n+1,118: POKE n+2,250
45 POKE 64050,118
50 SAVE "USR63637"CODE 63637,1857: VERIFY ""CODE: STOP
60 LET n=64030: GO SUB 70: LET n=64044
70 POKE n,253: POKE n+1,126: POKE n+2,2: RETURN
```

5. WHAM! THE MUSIC BOX

A Melbourne House jóvoltából 1985 óta a zeneszerető Spectrumosok új zenetervező programmal lettek gazdagabbak. Sajnos ezen a területen igen szegényes a választék, bár a BEEP néha 'csodákra' is képes, nem sok olyan felhasználói software áll rendelkezésre, amely kihasználná a BEEP gépi kódú összes lehetőségét. A 'MUSICMAKER' (Bellflower) csak 'ugatta' a témát, a 'MUSIC TYPEWRITER' (Romantic Robot) jó színvonalú, de inkább a zenetanulást segíti. A 'Spectrofon' (Andromeda) kiváló, de többszólamú zene készítését ez sem teszi lehetővé. A 'WHAM' hiánypótló volt, és ebben a kategóriában ma is az elsők között verseng.

A program betöltése után a főmenü jelentkezik be, melynek pontjait most tekintsük át részletesen:

1. LOAD TUNE

Ebben az opcióban a WHAM!-ből előzőleg kimentett zenét lehet betölteni a zenepufferba. Az opció három részre bontódik. Magnetofonról való betöltéskor a gép bekéri a zeneadatok nevét, de elég itt csak ENTER-t nyomni, ezesetben az első file-t tölti be a magnetofonról. Ha a memóriából akarunk tölteni, megjelenik egy lista, amely az öt beépített zene címét tartalmazza. A hatodik helyre, amely most üres, mi definiálhatunk zenét. Választáskor meg kell adnunk a zene azonosító számát. Microdrive-ról való töltéskor a gép először megkérdezi, hogy kérünk-e katalógust, vagy nem. Ezután a zene címét kell megadnunk. Itt nem elegendő csak ENTER-t nyomnunk! A SPACE segítségével visszatérhetünk a főmenübe. A zenepufferben lévő zenét tetszőlegesen módosíthatjuk.

2. SAVE TUNE

Ez az opció lehetővé teszi, hogy a zenepufferben lévő zenét kimentsük magnetofonra, microdrive-ra, vagy a memóriába. A menü meghívása után a LOAD-nál leírtaknak megfelelően kell eljárni. Ha a zenét a memóriában tároltuk el, az tetszés szerint felülírható.

3. HEAR TUNE

Ha meghívjuk ezt az opciót, a program lejátsza nekünk a zenepufferben található zenét.

4. WHAMPILER

Ezt a menüpontot csak akkor lehet meghívni, ha szerkesztéskor a zene végén a végjelzőt (SET LOOP) elhelyeztük (erről az EDITOR funkció tárgyalásánál bővebben szólunk).

Ez az egyik leghasznosabb funkció. A zenepufferben lévő zene assemblálására szolgál. Igen hasznos lehet saját játékprogramjaink zenekíséretéhez, melyekben kétszólamú zenét szeretnénk betétként elhelyezni. Az assemblálás a következőképpen történik: Attól a címtől, amit a kérdésre megadtunk, elhelyez a program egy kb. 400 byte hosszú gépi kódú programot, majd hangonként 2 byte-ot. Így pl. ha a zenénk hossza 100 hang, akkor a rutin után 200 byte kerül elhelyezésre, azaz összesen a kimentett gépi kódú program hossza kb. 600 byte lesz.

Bejelentkezéskor először meg kell adnunk az assemblálás kezdőcímét, amely nem lehet 32768-nál kisebb értékű. Ez a megkötés azért lett beépítve, mert az ULA minden 1/50-ed másodpercben levizsgálja a képernyő RAM-ot, és ez sajnos lassítja az alsó 16K RAM működését, ami itt problémákat okozhat. Ezután meg kell adnunk, hogy milyen néven mentse majd ki az adatokat a magnetofonra, vagy microdrive-ra. Utoljára a visszatérés módját kell beállítanunk.

1-KEYPRESS: Ha vége a zenének, kezdi újból lejátszani, egészen addig, amíg meg nem nyomunk egy billentyűt.

2-ALWAYS: Visszatér minden hang lejátszása után.

3-TUNE END: Vagy akkor tér vissza, ha vége a zenének, vagy akkor, ha közben megnyomunk egy billentyűt.

A visszatérésről tudni kell, hogy ilyenkor a program egy gépi kódú RET utasítást hajt végre, tehát az assemblált programunk beépíthető egy gépi kódú programba is. Kimentéskor a képernyőn a következő adatok jelennek meg:

ASSEMBLY ADDRESS: Az assemblálás kezdőcíme.
RETURN OPTION: A visszatérés módja.
WHITE NOISE: Fehér zaj lett elhelyezve a programban.
CHANNEL 1 LENGTH: Az 1.hangcsatorna által tartalmazott hangok száma.
CHANNEL 1 LOOP: Az 1.hangcsatornán van-e végjelző pont.

CHANNEL 2 LENGTH: A 2.hangcsatorna által tartalmazott hangok száma.
CHANNEL 2 LOOP: A 2.hangcsatornán van-e végjelző pont.
CODE LENGTH: Az assemblált gépi kódú rutin hossza byte-okban.
REPLAY SPEED: A lejátszási sebesség állításának a címe.
BORDER COLOUR: A lejátszás alatt a keret színének állítási címe.
TO RUN: A gépi kódú program indításának a címe.

FONTOS! A kimentett gépi kódú rutin futtatásához a WHAM!-nek nem kell a memóriában lennie.

5. SET TEMPO

Ebben az opcióban a zene lejátszási sebességét állíthatjuk be. A sebességet az '5' billentyűvel csökkenthetjük, illetve a '8' billentyűvel növelhetjük.

6. EDIT MODE

Itt van lehetőségünk a zene meg- ill. átszerkesztésére. Szerkesztéskor a képernyő három részre oszlik. Legfelül a kottapapír látható, erre íródik rá az éppen beírt hang. Ez nagyon jó pl. a zene javításakor vagy dokumentálásakor. A kotta alatt két sorban a legfontosabb információkat láthatjuk:

MODE: Állapota kétféle lehet. EDIT-ha a zenét szerkesztjük, ill. REPLAY-ha a zenét visszajátszuk.
OCTAVE: Oktáv mutató. Értéke 1-4-ig terjedhet, a kettes érték jelenti a C-t.
CHANNEL: Csatorna jelző. Értéke 1 vagy 2, attól függően, hogy melyik hangcsatornát használjuk.
COUNTERS: Hang számlálók. Azt adják meg, hogy az egyes vagy a kettes csatornán hányadik hangnál járunk.

Végül legalul egy zongora billentyűsora látható, azt mutatja, hogy a zongorán melyik hangot ütjük le. A képernyő után térjünk át a billentyűzetre.

1-4: oktáv váltók, ezekkel a billentyűkkel állíthatjuk be a kívánt oktáv számát.
5: A keret (BORDER) színének beállítására szolgál.
6: Visszatérés a főmenübe.
7: A zenepuffer törlésére szolgál. Törlés előtt megerősítést kér, biztos, hogy törölni akarjuk-e a puffert.
8: Megnyomásakor kikerülünk az EDITOR üzemmódból, és átkerülünk a fehér zaj szerkesztőbe. Ilyenkor a képernyőn két sor látható, a felsőben a zaj 'darabosságát' lehet beállítani (ezt úgy kell elképzelni, hogy az effektet négy részre osztjuk, és azt állítjuk be, hogy ebből a négy részből hány szólaljon meg), az alsóban pedig a hang mélységét szabályozhatjuk. Mindkét sor három különböző részre oszlik, amelyek az 'Y,U, I' billentyűkhöz tartoznak, tehát egyszerre háromféle fehér-zajt használhatunk a zenénkben. Az állító kurzort a kurzor-nyilakkal irányíthatjuk, a nullával állíthatjuk, a '9' billentyűvel pedig megszólaltathatjuk az aktuális effektet. Ha ENTER-t nyomunk, visszatérünk a zeneszerkesztőbe.
9: A kottapapírt előretekercseli.
0: (zérus) Törli a legutoljára beírt hangot.
Q: Az aktuális hangtól lejátsza a zenét.
W: Vég- jelző pontot tesz az aktuális helyre. Itt vége van a zenének. Fontos, hogy csak az egyik csatorna lezárására szolgál, a másik csatornán is be kell állítani ezt a pontot (lehetőleg ugyanoda).
E: Egy dobütésszerű hangot szólaltat meg.
R: Megnyomásakor a gép a kotta elejére áll, és a számlálókat nullázza.
Y,U,I: Ezekkel a billentyűkkel szólaltathatjuk meg a fehér zaj effekteket.
O: Hátratekercseli a kottapapírt.
P: A soron következő hangot szólaltatja meg.
A,S,D,F,G,H,K,L: A félhangok billentyűi (a zongorán a fekete billentyűk).
ENTER: Szünet-jel, a kottán a program R-rel jelöli.
CAPS SHIFT-SPACE-ig: Egész hangok.

Megjegyzés: A program csak ti-ket használ, így ha tá hangra van szükségünk, két ti-t kell beírnunk.

7. HELP PAGE

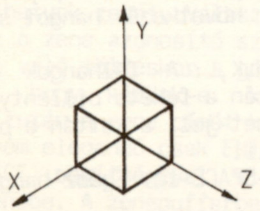
A szerkesztőben használt billentyűk rövid emlékeztetőjét jeleníti meg.

FONTOS! Ha bármilyen okból visszakerülnénk a BASIC-be, a program GO TO 0-val újraindítható.

Reméljük, hogy ez a tájékoztató sok felhasználó számára indíttatást ad majd a WHAM! rutinjainak saját játékprogramokba való beépítésére.

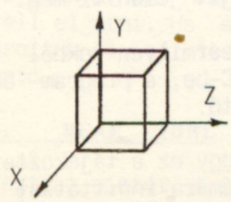
Egy 3D kép alapvetően 3D koordinátarendszerben építhető fel. A 3D tárgynak egyidőben látszik hosszúsága, szélessége és a magassága is. A képernyőn, úgy mint a fényképeken, csak két dimenzió létezik, de lehetőség van egyszerű programozástechnikai fogásokkal szimulálni a 3D képet a 2D képernyőn.
Először is azt kell ismerni, hogyan reprezentálunk egy képet három dimenzióban. Vegyünk elő egy papírt, rajzoljunk rá egy két tengelyből álló x-y koordináta rendszert, de úgy, hogy az origó a papír balalsó sarkába essen. Az x tengely jobbra, az y pedig felfelé mutat. Ebben a koordináta rendszerben minden egyes pont két koordinátával írható le (x,y). Ez gyakorlatilag a kétdimenziós koordináta geometria. A Spectrum PLOT utasítása is ennek megfelelően működik. Most vezessük be a harmadik tengelyt, mely úgyszintén az előbbi origóból indul ki, de jelen esetben csak egy pontnak látszik, és iránya a papír síkja felé mutat. Ez a tengely a z tengely. A 3D kép megmutatása a koordináta rendszer jelenlegi állásában nehézkes, így a látószöget általában átállítjuk, vagy úgy is mondhatjuk, a 3D koordináta rendszert térben elforgatjuk. A számítógépes gyakorlatban a 3D ábrázolás számos módszere ismert, most csak a fontosabb perspektívákat tekintjük itt át:

a/ Izometrikus (éltartó) perspektíva



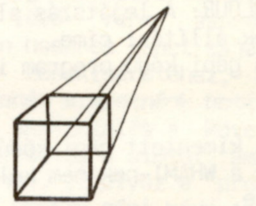
Ebben az ábrázolási módban az egymástól 120 fokban elhelyezkedő tengelyekre valóságos méreteket mérünk fel, perspektívikus rövidülés, torzulás nincs.

b/ Axonometrikus (területtartó) perspektíva



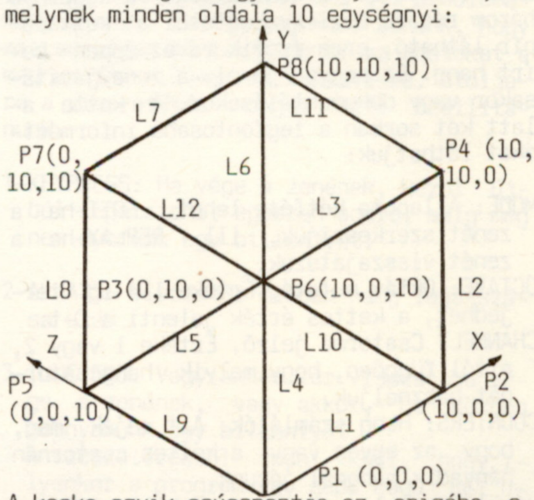
Ebben az ábrázolási módban a két egymásra merőleges koordináta tengely által határolt terület a 2D képnek megfelelő, ezért területtartó, a harmadik tengely ezektől 135 fokra helyezkedik el, és a valóságos méret felét mérjük fel rá.

c/ Iránypont perspektíva



Ebben az ábrázolási módban egy térben kijelölt ponthoz igazodik a teljes perspektíva.

A 3D koordináta rendszerben már minden egyes pont három koordinátával írható le (x,y,z).
Példánk további részét tekintsük meg az izometrikus ábrázolás szabályainak megfelelően, a koordináta rendszert pedig forgassuk el 180 fokkal. A koordináta rendszerbe vázoljunk egy szabályos kockát, amelynek minden oldala 10 egységnyi:



A kocka egyik csúcspontja az origóba esik, innen pedig minden főtengely irányába 10 egység kiterjedésű.
Példánkban a testet a következő pontok írják le:

P1(0,0,0)	P2(10,0,0)	P3(0,10,0)
P4(10,10,0)	P5(0,0,10)	P6(10,0,10)
P7(0,10,10)	P8(10,10,10)	

Két dimenzióban nincs lehetőség ezekkel a koordináta értékekkel felrajzolni a 3D pontokat, ezért szükség van egy segédprogramra, amely a 3D pontokat átkonver-

tálja 2D pontokká, szimulálva ezzel a 3D képet.
Ismernünk kell azt is, hogy a test milyen pontok kapcsolatából épül fel, vagyis azt, hogy melyik pontokat kell összekötni egymással, a test kialakításához. Térjünk vissza ismét a már megismert példára. Minden csúcspontot elneveztünk egy P betűből és egy számból álló azonosítóval (P1-P8), ugyanezt tettük az élekkel is, itt az elnevezés egy L betű és egy szám (L1-L12).

Készítsük el a 3D-2D konvertáló programot, amely BASIC-ben is megoldható:

```
 10 DIM P(8,3)
 20 DIM L(12,2)
 30 FOR I=0 TO 1
 40 FOR J=0 TO 1
 50 FOR K=0 TO 1
 60 LET P(4*I+2*J+K+1,1)=10*K
 70 LET P(4*I+2*J+K+1,2)=10*J
 80 LET P(4*I+2*J+K+1,3)=10*I
 90 NEXT K
100 NEXT J
110 NEXT I
120 FOR I=1 TO 12
130 FOR J=1 TO 2
140 READ L(I,J)
150 NEXT J
160 NEXT I
170 DATA 1,2,2,4,4,3,3,1
180 DATA 5,6,6,8,8,7,7,5
190 DATA 1,5,2,6,4,8,3,7
200 FOR I=1 TO 12
210 LET A=1: GO SUB 500
220 LET P1=5*P+128
230 LET Q1=5*Q+88
240 PLOT P1,Q1
250 LET A=2: GO SUB 500
260 DRAW 5*P+128-P1,5*Q+88-Q1
270 NEXT I
280 STOP
500 LET A=L(I,A)
510 LET X=P(A,1)
520 LET Y=P(A,2)
530 LET Z=P(A,3)
540 LET P=SQR (3)*(Y-X)/2
550 LET Q=Z-(Y+X)/2
560 RETURN
```

A 10. sorban dimenzionálunk egy tömböt, amely majd a csúcspontok koordinátáit fogja tartalmazni (8 csúcspont - 3 koordináta érték). A 20. sorban az éleknek biztosítunk egy tömböt (12 él - 2 pont között). A 30.-110. sorok inicializálják a P() tömböt, majd megtörténik a 10 egységnyi kocka csúcspont adatainak a tárolása. A 120.-190. sorok az L() tömböt töltik be a megfelelő adatokkal. Itt megjegyeznénk, hogy erre a célra a legegyszerűbb megoldás a DATA sorok vagy az INPUT alkalmazása.
Ezt követi a program szíve, a kocka felrajzolása.
Feltételezünk egy pontot a 3D koordináta rendszerben (x,y,z), hozzárendelünk egy pozíciót a képernyőről (p,q), amelyet a PLOT utasítás mellé megadhatunk. Ezután már csak egy reális számítási mechanizmust kell kidolgoznunk, amely az x,y,z koordinátákat átkonvertálja érvényes p,q pozíciókká.
A 3D koordináta rendszer 2D-ra való átkonvertálására számos módszer ismeretes, most itt csak egy megoldás alapelvét ismertetjük, az izometrikus perspektíváét. A módszerben két alapképlet szükséges p és q meghatározásához:

```
LET P = SQR (3) * (Y-X) / 2
LET Q = Z - (Y+X) / 2
```

Azok részére, akik egy kicsit szeretnék gyakorolni a gépi kódú programozást, javasoljuk a számítási mechanizmus gépi kódú megfelelőjének áttekintését és alkalmazását:

```
40000  42,75,92      LD   HL,(23627)
;HL-ben a változóterület mutatója
40003  126           LD   A,(HL)
;A = következő változó byte
40004  230,127       AND  127
;A 7. bit feleslegés
40006  55            SCF
40007  200           RET  Z
;Visszatérés, ha a változót nem találta
40008  185           CP   C
40009  200           RET  Z
;Visszatérés, ha a változót nem találta
40010  197           PUSH BC
;A keresett változónév tárolása
40011  205,184,25    CALL 6584
;ROM 19B8 hívása - DE-ben a következő
;változó mutatója
40014  235           EX   DE,HL
;A mutatót HL veszi fel
40015  193           POP  BC
;C-ben a változónév kódja
40016  24,241        JR   40003
;Ugrás vissza, keresés tovább
40018  14,122        LD   C,122
;A "z" változó kódja
40020  205,94,156    CALL 40030
;Ugrás a kereső szubrutinra
```

```
40023  14,121           LD   C,121
;Az "y" változó kódja
40025  205,94,156       CALL 40030
;Ugrás a kereső szubrutinra
40028  14,120           LD   C,120
;Az "x" változó kódja
40030  205,64,156       CALL 40000
;Keresés
40033  218,202,27       JP   C,7114
;ROM 1C2E hívása - hibaüzenet, ha
;nem találja a változót
40036  35               INC  HL
;HL - változótartalom mutató
40037  195,180,51       JP   13236
;ROM 33B4 hívása - változótartalom a
;kalkulátor verembe, majd visszatérés
40040  205,87,156       CALL 40023
;Y és X a kalkulátor veremben
40043  239              RST  40
;A verem tetején: Y,X
40044  3                DEFB=SUBTRACT
;Y-X
40045  52,64,176,0,3    DEFB=STK.DATA
;Y-X,3
40050  40               DEFB=SQR
;Y-X,SQR(3)
40051  4                DEFB=MULTIPLY
;SQR(3)*(Y-X)
40052  162              DEFB=CONST.HALF
;SQR(3)*(Y-X),1/2
40053  4                DEFB=MULTIPLY
;SQR(3)*(Y-X)/2
40054  56               DEFB=ENDCALC
;Műveletek vége
40055  14,112           LD   C,112
;A "p" változó kódja
40057  205,135,156      CALL 40071
;LET P=SQR(3)x(Y-X)/2
40060  205,82,156       CALL 40018
;z,y,x a kalkulátor veremben
40063  239              RST  40
;A verem tetején: Z,Y,X
40064  15               DEFB=ADD
;Z,Y+X
40065  162              DEFB=CONST.HALF
;Z,Y+X,1/2
40066  4                DEFB=MULTIPLY
;Z,(Y+X)/2
40067  3                DEFB=SUBTRACT
;Z-(Y+X)/2
40068  56               DEFB=ENDCALC
;Műveletek vége
40069  14,113           LD   C,113
;A "q" változó kódja
40071  205,64,156       CALL 40000
;A változó keresése
40074  48,10            JR   NC,40086
;Ugrás, ha létezik
40076  197              PUSH BC
;A változó kódjának tárolása
40077  1,6,0            LD   BC,0
;Helybiztosítás a
40080  205,85,22        CALL 5717
;változónak (ROM 1655 HEX)
40083  35               INC  HL
;HL - új hely mutatója
40084  193              POP  BC
;C - változónév kódja
40085  113              LD   (HL),C
;Tárolás
40086  35               INC  HL
;HL - változótartalom
40087  229              PUSH HL
;Tárolás
40088  205,191,53       CALL 13759
;ROM 35BF hívása - HL-ben a verem
;utolsó elemének a címe
40091  34,101,92        LD   (23653),HL
;Az elem törlése
40094  209              POP  DE
;DE - változó tartalma
40095  1,5,0            LD   BC,5
;Hossz beállítása
40098  237,176          LDIR
;A szám bemásolása a változóba
40100  201              RET
;Visszatérés
```

Ha vállalkozunk a gépi kódú rutin értelmezésére és bevitelére, a Basic programunkat módosítsuk. Töröljük az 540. és az 550. programsorokat, majd gépeljük be:

```
540 RANDOMIZE USR 40040
```

A gépi kódú rutin a megadott x,y,z koordináták alapján kiszámítja a p,q értékpárokat, és ezeknek megfelelően generál változót a változóterületen. Ezután már a BASIC program probléma nélkül el tudja végezni a pontok kijelölését. Természetesen merészebbek a teljes programot átírhatják gépi kóddá.

Ha programunkban megváltoztatjuk a 60.-80. programsorokat, pl. 6*K, 14*J, 17*I-re, nem kockát, hanem téglatestet kapunk eredményül. Próbáljuk meg megváltoztatni a koordináta értékeket és ismét nézzük meg az eredményt.

Az itt ismertetett mintapélda igen egyszerű és a kép felrajzolásán túl másra nem is képes. A továbbiakban olyan gyakorlati software-ekkel foglalkozunk, amelyek jóval bonyolultabb algoritmusokat használnak, és kiegészítik a rajzoló mechanizmust nagyon hasznos plusz szolgáltatásokkal.

Az OTTO motor működése

```
   5 REM "Otto"
   9 BRIGHT 1: CLS
  10 LET f=0  LET p=0: LET e=0: 
LET h=75: LET r=26
  20 LET k=1: LET s=1: LET sz=0:
 LET z=0
  30 PLOT 105,75: DRAW 0,82: DRA
W 44,0: DRAW 0,-82
  40 FOR y=145 TO 85 STEP -5: PL
OT 75,y: DRAW 30,0: NEXT y
  50 FOR y=145 TO 85 STEP -5: PL
OT 149,y: DRAW 30,0: NEXT y
  55 FOR y=155 TO 135 STEP -5: P
LOT 65,y: DRAW 40,0: NEXT y: FOR
 y=155 TO 135 STEP -5: PLOT 149,
y: DRAW 40,0: NEXT y
  56 FOR y=170 TO 160 STEP -5: P
LOT 65,y: DRAW 124,0: NEXT y
  57 PLOT 119,157: DRAW 0,5: DRA
W 16,0: DRAW 0,-5: PLOT INVERSE
1;120,157: DRAW INVERSE 1;14,0:
PLOT INVERSE 1;120,160: DRAW INV
ERSE 1;14,0
  58 PLOT 122,162: DRAW 0,8: PLO
T 132,162: DRAW 0,8: PLOT INVERS
E 1;123,165: DRAW INVERSE 1;8,0
  59 PLOT 118,170: DRAW 0,4: DRA
W 16,0: DRAW 0,-4: PLOT 127,162:
 DRAW 0,-1: PLOT 130,162: DRAW 0
,-2: DRAW -2,0
  60 PLOT 105,75: DRAW 44,0,2*PI
*(4.09/5)
  70 FOR x=107 TO 115: PLOT INVE
RSE 1;x,157: DRAW INVERSE 1;0,13
: NEXT x: PLOT 107,157: DRAW 0,1
3: PLOT 115,157: DRAW 0.13
  75 FOR x=147 TO 139 STEP -1: P
LOT INVERSE 1;x,157: DRAW INVERS
E 1;0,13: NEXT x: PLOT 139,157:
DRAW 0,13: PLOT 147,157: DRAW 0,
13
  90 LET y=125
 100 PLOT 107,y: DRAW 0,30: DRAW
 40,0: DRAW 0,-30
 101 LET w=r*SIN (p/180*PI)/h: L
ET f3=(ASN w)*180/PI
 102 LET f1=(180-(p+f3))+180+p
 103 LET f1x=SIN (f1/180*PI)*h:
LET f1y=COS (f1/180*PI)*h
 110 LET fx=SIN (f/180*PI)*r: LE
T fy=COS (f/180*PI)*r
 120 PLOT 127,40: DRAW fx,fy: DR
AW f1x,f1y: PAUSE 10
 129 IF s=1 THEN GO TO 131
 130 IF e>=180 THEN PLOT INVERSE
 1;107,147: DRAW INVERSE 1;8,0:
PLOT INVERSE 1;111,147: DRAW INV
ERSE 1;0,15: PLOT 111,157: DRAW
0,15: PLOT 107,157: DRAW 8,0: LE
T s=1
 131 IF e<180 THEN PRINT AT 20,1
;" SZIV  "
 132 IF e>=180 AND e<360 THEN PR
INT AT 20,1;" SURIT "
 133 IF e>=360 AND e<540 THEN PR
INT AT 20,1;" ROBBAN"
 135 IF e>=540 AND e<=720 THEN P
RINT AT 20,1;"KIPUFOG"
 139 IF sz=1 THEN GO TO 141
 140 IF e<540 OR e>=680 THEN PLO
T INVERSE 1;139,147: DRAW INVERS
E 1;8,0: PLOT INVERSE 1;143,147:
 DRAW INVERSE 1;0,15: PLOT 139,1
57: DRAW 8,0: PLOT 143,157: DRAW
 0,15: LET sz=1
 200 PLOT INVERSE 1;107,y: DRAW
OVER 1;0,30: DRAW OVER 1;40,0: D
RAW OVER 1;0,-30
 201 IF y>=125 THEN LET k=1
 202 IF k=0 THEN GO TO 211
 210 LET y=y-5
 211 IF y=70 THEN LET k=0: LET y
=y+5
 212 IF k=1 THEN GO TO 221
 220 LET y=y+5
 230 PLOT 127,40: DRAW OVER 1;fx
,fy: DRAW INVERSE 1;f1x,f1y
 240 LET f=f+18: LET p=p+18: LET
 e=e+18
 250 IF f>=360 THEN LET f=0
 255 IF p=360 THEN LET p=0
 256 IF e>=720 THEN LET e=0
 260 IF e<10 THEN PRINT AT 1,1;"
00";e
 265 IF e>10 AND e<100 THEN PRIN
T AT 1,1;0;e
 270 IF e>=100 THEN PRINT AT 1,1
;e
 279 IF s=0 THEN GO TO 281
 280 IF e>18 AND e<180 THEN PLOT
 INVERSE 1;107,157: DRAW INVERSE
 1;8,0: PLOT INVERSE 1;111,157:
DRAW INVERSE 1;0,15: PLOT 107,14
7: DRAW 8,0: PLOT 111,147: DRAW
0,15: LET s=0
 289 IF sz=0 THEN GO TO 291
 290 IF e>=540 AND e<680 THEN PL
OT INVERSE 1;139,157: DRAW INVER
SE 1;8,0: PLOT INVERSE 1;143,157
: DRAW INVERSE 1;0,15: PLOT 139,
147: DRAW 8,0: PLOT 143,147: DRA
W 0,15: LET sz=0
 299 IF z=1 THEN GO TO 301
 300 IF e>360 AND e<522 THEN PRI
NT FLASH 1;AT 2,15;"*": FOR n=0
TO 10: BEEP .01,-10: NEXT n: LET
 z=1
 309 IF z=0 THEN GO TO 311
 310 IF e>522 THEN PRINT AT 2,15
;" ": LET z=0
 400 GO TO 100
```

Az itt közölt BASIC program elsősorban azoknak a Spectrumosoknak lehet hasznos, akik a számítástechnika mellett egyéb műszaki ismeretek felé is orientálódnak. A program szemléletesen bemutatja a négyütemű OTTÓ motor működését, különös tekintettel az egyes ütemek (szívás, sűrítés, munkaütem, kipufogás) fázisainak megjelenítésére. A fázisrajzok jól ábrázolják a dugattyúk, a hajtókar, valamint a szelepek ütemtől függő állapotát, ezáltal hamar elsajátítható a rendszer működése. A program oktatási célokra kiválóan alkalmas.



A "PLAY" utasítás

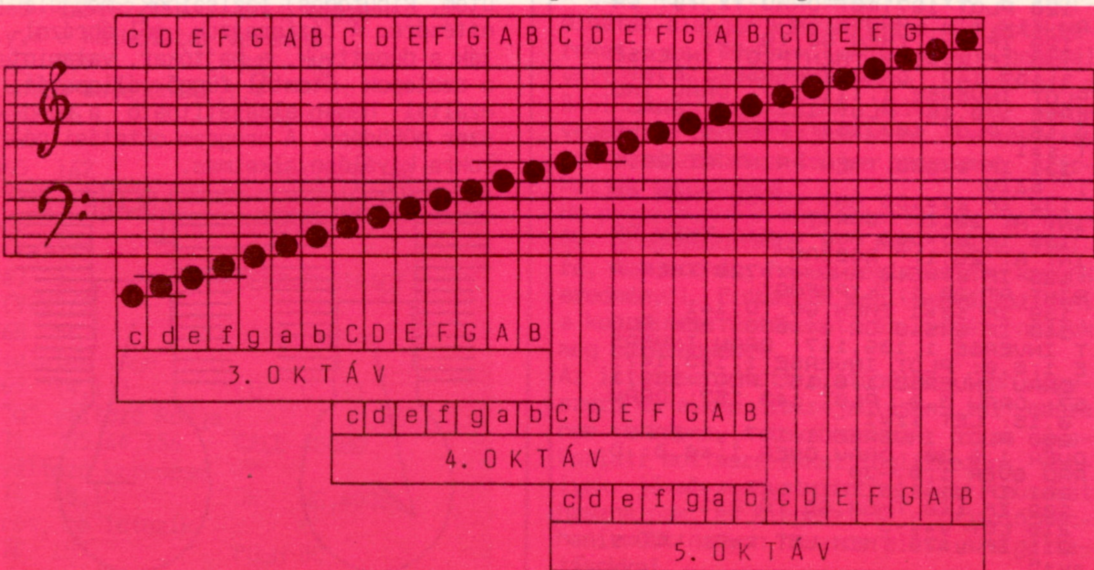
A PLAY sokkal rugalmasabb a BEEP-nél, a három egymástól független hang segítségével a harmóniák és effektek igen széles arzenálja vonultatható fel. Használata rendkívül egyszerű.

Pl. szólaltassuk meg az "A" hangot az egyvonalas "C" alatt fél másodpercig: PLAY "A"
Most játsszunk egy C major skálát: PLAY "cdefgahC"

Megjegyeznénk, hogy az utolsó "C" nagy betű. Ez azt jelzi a PLAY utasításon belül, hogy egy oktávval magasabban szólaltassa meg, mint a kis "c" betűvel jelzett hangot. Az itt lejátszott skála tehát megfelel egy oktávnyi hangterjedelemnek. Az itt látható mintapéldát 'C major' skálának hívjuk, melynek érdekessége, hogy az egyetlen olyan major (dúr) skála, amelynek zongorával történő megszólaltatásához egyetlen fekete billentyű használata sem szükséges.
Két alapskálát különböztetünk meg, a majort és a minort.
Most játsszuk le a 'c minor' skálát: PLAY "cdefga$bC"
Bevezettünk egy újabb jelet, a '$'-t. Ez fél hanggal lejjebb szállítja a mögötte álló hangot. A zenei életben "b" előjegyzésnek felel meg. Használhatjuk a PLAY utasításban a "#" előjegyzést is, amely egy fél hanggal feljebb szállítja a hangot, s ezt a valóságban is így jelöljük.
A PLAY utasítás 9 teljes oktávot (9-szer 7 hangot) képes átfogni. Az oktávok közötti váltás az "O" (nagybetű) és az azt követő szám (0-8) megadásával érhető el. Most tekintsük meg a következő programot:

```
10 LET o$="O5": LET n$="DECcg": LET a$=o$+n$
20 PLAY a$
```

Ez újabb lehetőséget tár elénk.Nem szükséges mindent a PLAY mellett megadnunk. A vezérlést stringbe is tehetjük, a stringeket összegezhetjük, s a PLAY segítségével mint egy stringre is hivatkozhatunk. Az itt látható kis program megfelel a következőnek: PLAY "O5DECcg"
Tekintetbe véve, hogy a PLAY nem korlátozza a string hosszát, viszont egy hosszabb string kezelése bonyolultabb lehet, így célszerűbb több rövidebb string összegzésével előállítani a célstringet. Változtassuk meg a 10. sorban az 'O5'-öt 'O7'-re, és futtassuk ismét a programot, ezután nézzük meg, mit eredményez az 'O2'. Ha nem definiáljuk az oktáv számát, azt a PLAY alapértelmezésben 5-nek veszi.
A PLAY további elemzése előtt tekintsük meg a zenei skála-diagramot:



Láthatjuk, hogy a PLAY oktávjai átfedik egymást, pl. 'O3D' ugyanazt jelenti, mint 'O4d'. Ez megkönnyíti a dolgunkat, mivel nem kell olyan sűrűn oktávot váltani, mintha oktávonként csak egy skálát programozhatnánk. Néhány hang a legalsó oktávban egyszerűen nem szólal meg, mivel arra technikailag sem nyílt lehetőség. A PLAY segítségével beállítható az egyes hangjegyek megszólalási ideje (időtartama) is. Ez egyszerűen egy egyedülálló számjegy segítségével választható ki. Változtassuk meg a 10. sort: 10 LET o$="2" és futtassuk ismét a programot. Ezután próbáljuk ezt az értéket megváltoztatni 1-9 között.

Megjegyzés: Ezek a konstansok nem konkrét időegységeket, csak arányokat állítanak be.

Most tekintsük át a konstansok jelentését:

1 tizenhatod	2 nyújtott tizenhatod	3 nyolcad
4 nyújtott nyolcad	5 negyed	6 nyújtott negyed
7 fél	8 nyújtott fél	9 egész

A PLAY képes háromszorozott hangok előállítására is:

10 3x tizenhatod	11 3x nyolcad	12 3x negyed

Azt az időt, amíg nincs megszólaló hang, szünetnek nevezzük, kottákon szünetjellel jelöljük, a PLAY utasításon belül "&" jelet kell alkalmazni. A szünet hossza mindig azonos az aktuális hangjegy hosszával. Pl. 10 LET o$="O4": LET n$="DEC&cg" C és c között szünetet eredményez.

Két hangjegy megszakítás nélküli összefűzését "tied notes"-nek hívjuk, jelölése a PLAY mellett egy "_" (aláhúzás) karakter. Így egy negyed és egy fél hang összevonható a következőképpen: "5_7c". A numerikus értékeket nem szabad kevernünk!

Pl: 10 LET o$="O62" jelentése: a 6. oktávban egy nyújtott tizenhatod, de ezt így nem fogadja el a PLAY (n: out of range hiba), mivel 'O' után egy numerikus értéket olvas, mely esetben a numerikus paramétereket el kell határolnunk egymástól egy "N" (nagybetű) karakter segítségével.

Helyesen tehát: 10 LET o$="O6N2"

A hangerő beállítható Ø és 15 (min-max) érték között, a "V" karakter és az azt követő számjegy segítségével. Megjegyezzük, hogy a 1Ø-15 közötti értékek használata ajánlatos, mivel a 0-9 viszonylag halkan szól.

Egynél több csatorna párhuzamos megszólaltatása igen egyszerű, a PLAY mellett vesszővel elhatárolva kell megkülönböztetni az előre definiált stringeket:

```
10 LET a$="O4cCcCgGgG": LET b$="O6CaCeSbdSbD": LET c$= "O5cCcCgGgG"
20 PLAY a$, b$, c$
```

A lejátszott zene sebessége ill. az egyes hangok hossza, a továbbiakban "tempó", csatornánként külön nem variálható, és kötelezően mindig az "A" csatorna stringjében (a PLAY utáni első string) kell megadni. Jele a "T" karakter ill. az ezt követő szám (60-240), alapértelmezése 120.

```
10 LET t$="T1ØØ": LET a$=t$+"O4cCcCgGgG": LET b$="O6CaCeSbdSbD"
20 LET c$="O5cCcCgGgG": PLAY a$, b$, c$
```

Majd futtassuk a programot különböző tempóval.

Jelentős szolgáltatása a PLAY utasításnak az egyes zenei részletek ismétlési lehetősége. Egy string bármely részlete ismételhető, ha azt zárójelek közé helyezzük: pl.

```
10 LET t$="T1ØØ": LET a$=t$+"O4(cC)(gG)"
20 LET c$="O5(cC)(gG)": PLAY a$, b$, c$
```

megfelel az előzőeknek.

Ha a string végére egy záró zárójelet teszünk (de nincs nyitó párja), úgy az adott string dallamát végtelen ideig ismételten játszani fogja. Ez igen hasznos pl. basszus vagy ritmus alapok bevezetéséhez. Pl. a PLAY "O4N2cdefgfed)" ciklusból csak a BREAK segítségével léphetünk ki. Most próbáljuk ki a következőt:

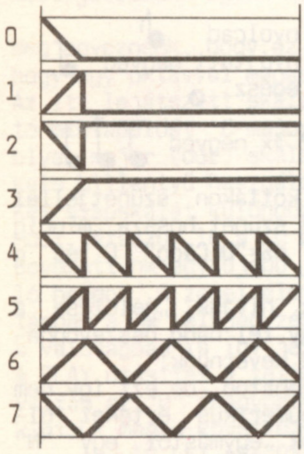
PLAY "O4N2cd(efgf)ed)". Vagyis a végtelen ütemben játszott basszus alapon belül is elhelyezhetünk ismétlődő részleteket. Most futtassuk a következő programot:

```
10 LET a$="cegbdfaC": LET b$="O4cC)": PLAY a$,b$
```

Futtatáskor azt tapasztaljuk, hogy a szóló befejezése után a basszus alap tovább szól, végtelen ciklusban. Nagyon hasznos dolog, ha a basszus alap is csak addig szól, amíg a főszólam. Erre a 'H' karakterrel nyílik lehetőség. Ha a 'H' karaktert bárhol elhelyezzük egy stringben ('H'=HALT=megállás), a PLAY ezt úgy értelmezi, hogy abban a pillanatban kapcsoljon ki minden hangot, amely éppen szól.

Mindenekelőtt változtassuk meg az a$ stringet: LET a$="cegbdfaCH" és futtassuk újra a programot.
A PLAY mellett beállíthatjuk az aktuális hullámformát is, a 'W' karakter segítségével, és az ezt követő numerikus értékkel (Ø-7). Mindemellett be is kell kapcsolnunk a burkológörbe-generátort, amelyet az 'U' karakterrel adhatunk meg egy stringben. Hasonlóképpen a 'V' (VOLUME=hangerő) és az azt követő számérték (Ø-15) a hangerőt állítja be, s mindegyik csak arra a csatornára vonatkozik, amelyik stringben elhelyeztük.

Az egyes burkológörbék:



Most próbáljuk ki a következőt:

```
10 LET a$="UX1ØØØWØC&W1C&W2C&W3C&W4C&W5C&W6C&W7C"
20 PLAY a$
```

Az 'U' bekapcsolja az effektet, a 'W' szelektálja a hullámformát, de látunk egy új jelölést is: 'X1ØØØ'.
Az 'X' melletti értékkel (Ø-65535) állítható be a burkológörbe-frekvencia, melynek alapértelmezése 1ØØØ. Próbáljuk megváltoztatni. A burkológörbe-frekvencia hatása csak a 4-7 hullámformák kiválasztása esetén hatásos. A PLAY parancs segítségével beállíthatjuk, hogy csatornánként tiszta hangot, fehér zajt, vagy e kettő keverékét szeretnénk előállítani. Ez az 'M' karakter és az ezt követő számérték (1-63) segítségével lehetséges. Nézzük meg használatához a beállító táblázatot:

	Hang csatornák			Zaj csatornák		
	A	B	C	A	B	C
érték	1	2	4	8	16	32

Ki kell választanunk, hogy mit szeretnénk, majd össze kell adni az ezeknek megfelelő számértékeket. Pl. ha az 'A' csatornán zajt, a 'B' csatornán tiszta hangot, a 'C' csatornán pedig mindkettőt szeretnénk, úgy a következő értékeket össze kell adni: 8 + 2 + 4 + 32 = 46.
Írhatunk bonyolultabb szimfóniát is, de ez esetben célszerű itt-ott megjegyzéseket tennünk a stringek belsejébe, a könnyebb eligazodás céljából. Pl. egy bonyolultabb program része lehet:

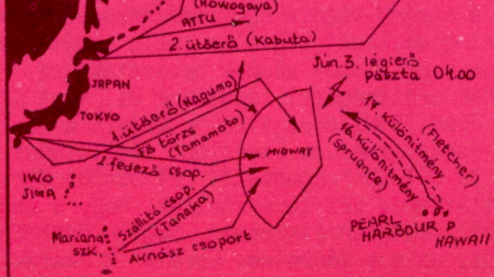
```
2045 LET z$=z$+"CDcE3Ge4_6f! a 24. tétel vége !egeA"
```

A megjegyzéseket mindig felkiáltójelek közé kell helyeznünk.

<u>MIDI PORT:</u> Ha rendelkezünk elektronikus zenei berendezéssel (pl. szintetizátorral, dobgéppel, vagy sequencer-rel), amely tartalmaz MIDI port-ot, úgy ezek a berendezések vezérelhetők a SINCLAIR SPECTRUM 128K gép segítségével. Ez megoldható a PLAY utasítás segítségével, és gépi kódú oldalról is. Az 'Y' karakter és az azt követő érték (1-16) a megfelelő zenei csatornát választja ki, míg a 'Z' karakter után adjuk meg az aktuális MIDI program kódot.

Battle for Midway

Elsősorban a hadi-stratégiai játékok kedvelőinek ajánljuk a PSS cég játékát, amely az amerikai-japán háború egy epizódját dolgozza fel 1942-ből. Június 4.-ét írunk; flottánk feladata, hogy a MIDWAY szigetek közelében összpontosuló japán flották ellen támadást intézzen, meghátrálásra kényszerítve az ellenséget. A játék során a japán csapatok kibontakozását háromféleképpen dolgozza fel a program. A május 25-28-i kiinduló álláspontot segítségképpen közöljük.



Japán haderő állása 1942.május 25-28.

- Regiszterpár-tartalom áttöltése a veremmutatóba

```
     249    F9     LD SP,HL   ;HL tartalma a veremmutatóba
 221,249    DD F9  LD SP,IX   ;IX tartalma a veremmutatóba
 253,249    FD F9  LD SP,IY   ;IY tartalma a veremmutatóba
```

Ezek speciális utasítások, melyekkel érdekes trükkök valósíthatók meg gépi kódú programjaink írása során. A gépi verem (machine stack) a BASIC terület végén található, mélysége változó, benne különféle - elsősorban átmeneti tárolásra alkalmas - adatokat helyezhetünk el. A legutoljára bevitt adat helyezkedik el a verem tetején, ezt a pozíciót (a verem tetejét) a processzorban mindenkor az SP regiszter (Stackpointer=veremmutató) jelzi. Egy gépi kódú program futásakor a veremmutató által jelölt címen, a verem tetején a mindenkori visszatérési cím tárolódik. Ha előzetesen más manipulációt nem végeztünk, úgy az SP regiszter tartalma 65344, itt pedig nem mást találunk, mint a BASIC INTERPRETER visszatérési címét, azaz ha SP-t a program futása közben nem változtatjuk meg, akkor a rutin végén található RET hatására a vezérlést a visszatérési cím szerint folytatja, vagyis visszatér a BASIC rendszerhez. A program futása közben ezt az értéket megváltoztathatjuk, s így RET után a vezérlés már a megadott címen folytatódik. Ezt a trükköt csak gyakorlottabb programozóknak ajánljuk, mert könnyen 'elszállhat' a rendszer!

4. Regiszterek töltése a memória adott rekeszéből

A memóriarekeszek és a processzor regiszterei közötti adatátvitel három jól elkülöníthető módon hajtható végre:

- abszolút címzéssel
- indirekt címzéssel

vagy - indexelt címzéssel.

Mielőtt áttekintenénk az egyes utasításokat, ismerkedjünk meg az itt említett fogalmakkal.

- Abszolút címzés: Az utasításban egy konkrétan megnevezett címre hivatkozunk, pl.: a 40000. cím tartalmát helyezze át a HL regiszterpárba.
- Indirekt címzés: Az utasításban egy közvetetten megadott címre hivatkozunk, pl.: a HL regiszterpár által meghatározott cím tartalmát helyezze át az A regiszterbe. Itt az adott (kiválasztott) cím értéke előzetesen HL-ben beállítható!
- Indexelt címzés: Az utasításban egy közvetetten megadott cím környezetére hivatkozunk, pl.: az IX regiszterpár által meghatározott cím feletti 50. cím (memóriarekesz) tartalmát helyezze át a B regiszterbe. Itt az adott báziscím értéke előzetesen IX-ben beállítható.

- Regiszterek töltése abszolút címzéssel

A jelölésekben 'nn' a decimális, 'NN' a hexadecimális cím, 'x' és 'y' a decimális, 'X' és 'Y' a hexadecimális cím alsó//felső byte-ja.

Pontosabban: x + (256 * y) = nn
YX egymás mellé írva: NN

Konkrét példával: 64 + (256 * 156) = 40000
mivel a 40000. cím alsó byte-ja = 64 (x)
felső byte-ja = 156 (y)
a 40000. cím hexadecimálisan = 9C40
így X = 40 és Y = 9C

```
      58,x,y     3A X Y    LD A,(nn)   ;az 'nn' cím tartalmának
      42,x,y     2A X Y    LD HL,(nn)  ;töltése a meghatározott
 237,107,x,y  ED 6B X Y    LD HL,(nn)  ;regiszterbe
 237, 75,x,y  ED 4B X Y    LD BC,(nn)
 237, 91,x,y  ED 5B X Y    LD DE,(nn)
 221, 42,x,y  DD 2A X Y    LD IX,(nn)
 253, 42,x,y  FD 2A X Y    LD IY,(nn)
 237,123,x,y  ED 7B X Y    LD SP,(nn)
```

Hamar szemünkbe ötlik, hogy a mnemonik-jelölésben ezek az utasítások csak annyiban különböznek a konstans adatok töltésétől, hogy <u>az adatot kerek zárójelek közé helyezzük</u>.

pl.: LD BC,40000 és LD BC,(40000)

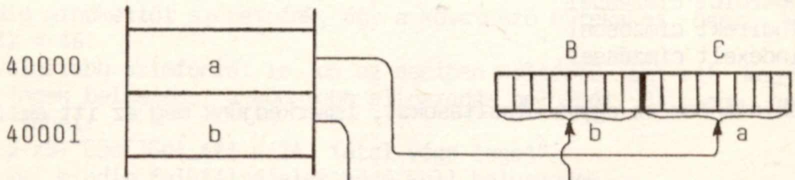
között az a különbség, hogy az első esetben BC-be 40000 íródik, míg a második esetben a 40000. cím tartalma.

A felsorolt 8 utasítás között, bármennyire is furcsa, kettőnek a hatása megegyezik, ugyanis az 'LD HL,(nn)' kiváltására két lehetőségünk is van. Mivel az első csak két byte-os utasítás, így célszerű annak a használata.

Ismét felvetődik egy kérdés. Mennyiben különbözik az 'A' regiszterbe történő töltés a többi regiszterpárba való töltéstől. Erre is nézzünk mintapéldát:

```
LD  A,(40000)
LD BC,(40000)
```

Az első esetben az 'A' regiszterbe beíródik a 40000. cím tartalma, míg a második esetben a 'C' regiszterbe beíródik a 40000. ill. a 'B' regiszterbe a 40001. cím tartalma, azaz 8 bites regiszterpárok és 16 bites regiszterek esetén attól függetlenül, hogy csak egy címet adtunk meg, az adott cím és az azt követő cím tartalma párhuzamosan helyeződik át, de jól jegyezzük meg, a báziscím (jelen esetben 40000) tartalma mindig az adott regiszterpár második betűjelével (jelen példánkban 'C') jelzett regiszterébe kerül !



Az utasításokra BASIC-ben a PEEK (memóriatartalom kiolvasása) a legjobb analógia. Előző példánk BASIC-ben a következőhöz hasonlítható:

```
10 LET C = PEEK 40000
20 LET B = PEEK 40001
30 LET BC = C + (256 * B)
```

Ezek az utasítások a jelzőbiteket nem befolyásolják !

- <u>Regiszterek töltése indirekt címzéssel</u>

Mechanizmusuk az előző csoporthoz hasonló, de az adott címek paramétereit mindig <u>előzetesen be kell állítani</u> a megfelelő regiszterekben. Előnyük, hogy egy byte-osak, hátrányuk, hogy csak néhány meghatározott regiszter ill. regiszterpár között hajthatók végre.

```
126   7E   LD A,(HL)   ;Az adott regiszterpárban
 10   0A   LD A,(BC)   ;meghatározott cím tartal-
 26   1A   LD A,(DE)   ;mának áttöltése egy
102   66   LD H,(HL)   ;regiszterbe
110   6E   LD L,(HL)
 70   46   LD B,(HL)
 78   4E   LD C,(HL)
 86   56   LD D,(HL)
 94   5E   LD E,(HL)
```

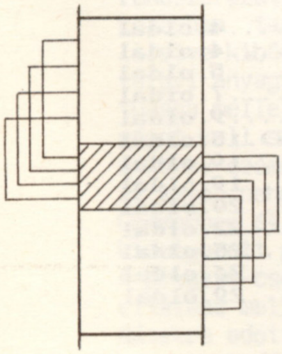
Gyakran gondot okozhat, hogy technikai okokból a címet más regiszterpárban, pl. a 'BC'-ben kell megadnunk, nem kell megijedni, a megoldás nagyon egyszerű.
Pl. a 40000. cím tartalmát az 'E' regiszterbe kellene tölteni, és a címet most BC tartalmazza.

```
LD H,B
LD L,C
LD E,(HL)
```

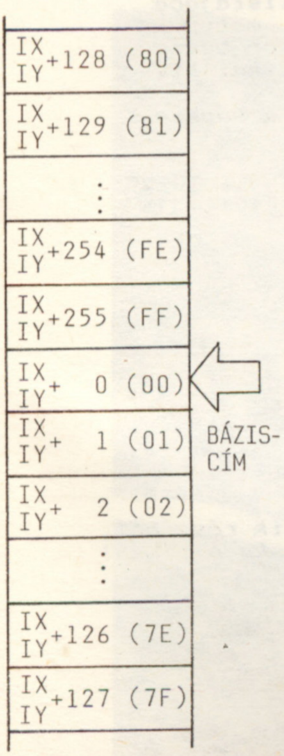
Ennek a kis trükknek a segítségével a művelet végrehajtható, ám a HL aktuális értéke felülíródik. Ilyen esetben HL-t átmenetileg célszerű eltárolni (a PUSH/POP utasításokat ld. később).

- Regiszterek töltése indexelt címzéssel

Az utasítás mechanizmusának megértéséhez egy példát szemléltetünk:



Gépi kódú programunk egy kiemelt szeletét látjuk, melyben sok-sok szubrutint helyeztünk el. Minden szubrutinnak más és más a funkciója (pl. egy játékprogramban a sprite-ok görgetése, háttérgrafika előhívása, stb). Ennek a megannyi szubrutinnak a startcímét célszerű egy külön un. ugró-táblázatban összegyűjteni (sraffozott terület), melynek báziscímét az IX, vagy IY regiszterekben tárolhatjuk el. Ezt követően az egyes szubrutinok indító címeit nem külön adatként kezeljük, hanem a báziscímektől való eltéréseikkel azonosítjuk őket.
Pl. IX + 12, az IX-ben megadott báziscímnél 12 byte-tal feljebb lévő címet határozza meg.




```
221,115,e    DD 73 E    LD A,(IX+e)    ;A báziscímmel és
221,102,e    DD 66 E    LD H,(IX+e)    ;eltolással megha-
221,110,e    DD 6E E    LD L,(IX+e)    ;tározott cím tar-
221, 70,e    DD 46 E    LD B,(IX+e)    ;talmát az adott
221, 78,e    DD 4E E    LD C,(IX+e)    ;regiszterbe, vagy
221, 86,e    DD 56 E    LD D,(IX+e)    ;regiszterpárba
221, 94,e    DD 5E E    LD E,(IX+e)    ;tölti.

253,115,e    FD 73 E    LD A,(IY+e)
253,102,e    FD 66 E    LD H,(IY+e)
253,110,e    FD 6E E    LD L,(IY+e)
253, 70,e    FD 46 E    LD B,(IY+e)
253, 78,e    FD 4E E    LD C,(IY+e)
253, 86,e    FD 56 E    LD D,(IY+e)
253, 94,e    FD 5E E    LD E,(IY+e)
```

A jelölésekben 'e' a decimális, 'E' a hexadecimális eltolási adatot jelzi.
Az eltolási adatok értelmezéséhez a bal oldalon látható egyszerű vázlatot készítettük.
Mint látható, a báziscímtől növekvő léptékben felfelé csak 128 értékkel léphetünk. Véletlenül sem célszerű eltéveszteni a dolgot, mert:
(IX ill. IY) + 128 = (IX ill. IY) - 127
vagy
(IX ill. IY) + 255 = (IX ill. IY) - 1
Ezért, ha táblázatunkban az adatok száma 128-nál kisebb, úgy a báziscímet előre érdemes felvenni, míg ha 128-nál nagyobb, akkor az adatmező közepére.

TARTALOMJEGYZÉK:



A második rész 25. oldalán közölt BASIC program listájába apró hiba csúszott, melynek következtében a 10. sor bevitelekor a gép szintaktikai hibát jelez.
A hibátlan programsor a következő:

```
10 FOR i=1 TO 10: FOR j=11 TO 20:
   PRINT AT i,j; CHR$ (40+i+j):
   NEXT j: NEXT i
```


A szerzők a következő címen érhetők el :
SPECTRUM VILÁG - BUDAPEST-3, *Postán maradó*, 1300

Akiknek nem jutott volna a 'SV' első ill. második részéből, azoknak azt ajánljuk, rendeljék meg a fent olvasható címen!

Felelős kiadó: Rucz Lajos, Székely László · K. F. eng.-sz.: 59406
10 000 – 807214 Pátria Nyomda
Felelős vezető: Vass Sándor vezérigazgató

LSI

LSI ALKALMAZÁSTECHNIKAI
TANÁCSADÓ SZOLGÁLAT

Postacím: BUDAPEST
POSTAFIÓK 121
1300

DR. OBÁDOVICS J. GYULA

MATEMATIKA

A könyv két kötete a középiskolai, technikumi matematika anyagot rendszerezett formában tárgyalja. Segítséget kíván nyújtani a korábban tanult ismeretek felfrissítéséhez, újratanulásához, illetve kibővítéséhez. A " Nyitott Egyetem " matematikai és szaktárgyi anyagának elsajátításához a könyv elméleti anyagának biztos tudása mellett, annak gyakorlati alkalmazásában is kellő jártasságot kell szerezni. Az Olvasót ebben való törekvésében a könyv számos kidolgozott példája támogatja.
Az anyag nehezebb részeinek elsajátítását a Commodore 64 típusú személyi számítógépre kidolgozott program-csomag is elősegíti.
A program-csomag az oktatási feladatok ellátása mellett az ellenörző kérdésekre adott válaszok alapján a tudásszint mérését is lehetővé teszi.



39,- Ft

LAPOZGATÓ SOROZAT

A Műszaki Könyvkiadó 1987 végén kezdte el a Lapozgató sorozat kiadását. A sorozatnak az a célja, hogy az IBM PC XT/AT-felhasználók olyan magyar nyelvű segédleteket kapjanak, amelyeket napi munkájuk során – egy-egy szoftver alkalmazásakor – szinte a számítógép tartozékaként, emlékeztetőként használhassanak. A kötetek az eredeti dokumentációra épülve megadják az utasítások, eljárások, függvények pontos szintaxisát, rövid leírását, és mindezt megjegyzésekkel, példákkal egészítik ki. Az utasítások funkció, ill. ábécé szerinti csoportosítása, valamint a könyvek spirálozott, zsebkönyv méretű kivitele a gyors információkeresést – és ami a fő, a gyors megtalálást – segíti.

A sorozat kötetei:

Herneczki Katalin: PC-DOS 3.20 (megj. 1987. december, ára: 120 Ft).
A legismertebb, legjobban elterjedt operációs rendszerrel foglalkozik.

•

Herneczki István: MACRO Assembler (terv. megj. 1988. február, ára: 150 Ft).
Az Intel 8088-as processzor regisztereit, címzési módjait, utasításkészletét, pszeudoutasításait írja le.

•

Szolek András: dBASE III PLUS (terv. megj. 1988. március, ára: 150 Ft).
Az Ashton-Tate által forgalmazott dBASE III PLUS relációs adatbáziskezelő rendszert foglalja össze.

•

Donát János: WordStar (terv. megj. 1988. április, ára: 120 Ft).
Az IBM PC számos szövegszerkesztője közül az egyik legelterjedtebb a WordStar, amely kedvelt segédeszköz levelek, cikkek, kézikönyvek készítése során.

•

Szolek András: Turbo Pascal (előkészületben).
Az egyik legnépszerűbb és legjobban használható programozási nyelvet, az amerikai Borland szoftverház termékét tárgyalja.

•

dr. Barakonyi Károly: FRAMEWORK II (előkészületben)
A FRAMEWORK II integrált programcsomag szövegszerkesztésre, táblázat- és adatbázis-kezelésre, grafikus feladatok megoldására, telekommunikációra egyaránt használható. Az összefüggő feladatokat funkcionálisan és vizuálisan is összekapcsolja.

A könyvek megrendelhetők:

MŰSZAKI KÖNYVKIADÓ Kandó Kálmán Könyvesboltja
Budapest, V. Bajcsy-Zs. út 20. –1051